НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

«До захисту допущено»
Завідувач кафедри
_____ Наталія АУШЕВА
“___”_____2026 р.

Дипломна робота
на здобуття ступеня бакалавр

За освітньою програмою “Цифрові технології в енергетиці”

Спеціальності 122 “Комп’ютерні науки”

на тему: «Система прогнозування інтенсивності атмосферних опадів на основі метеорологічних даних із використанням методів машинного навчання»

Виконала: студентка 4 курсу, групи ТР-22
Нагаївська Дарина Олександрівна _____
(прізвище, ім’я, по батькові) (підпис)

Керівник асистент каф. цифрових технологій____ ___
в енергетиці, Микита ГОЛОВАКІН _____
(посада, науковий ступінь, вчене звання, ім’я, ПРІЗВИЩЕ) (підпис)

Рецензент професор каф. теплової та альтернативної
енергетики, д.т.н., проф., Ольга ЧЕРНОУСЕНКО _____
(посада, науковий ступінь, вчене звання, ім’я, ПРІЗВИЩЕ) (підпис)

Н.контроль доцент каф. цифрових технологій_____
в енергетиці, д.ф., Артем ГУРІН _____
(посада, ім’я, ПРІЗВИЩЕ) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2026

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – перший (бакалаврський)
спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ
Завідувач кафедри ЦТЕ
_____ _Наталія АУШЕВА
(підпис)
“___” _____2026 р.

ЗАВДАННЯ
на дипломну роботу студенту

Нагаївській Дарині Олександрівні
(прізвище, ім’я, по батькові)

1. Тема роботи «Система прогнозування інтенсивності атмосферних опадів на основі метеорологічних даних із використанням методів машинного навчання»
Науковий керівник Головакін Микита Андрійович, асистент каф. ЦТЕ
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “28” травня 2026 року № 1992-с

2. Термін подання студентом роботи 04.06.2026

3. Вихідні дані до роботи: мова програмування Python 3.12, бібліотеки PyTorch 2.9.0, XGBoost 2.1, Pandas, NumPy, Scikit-learn, Streamlit, Plotly, середовище розробки Visual Studio Code.

4. Перелік питань, які потрібно розробити:

1) Провести аналіз існуючих методів прогнозування атмосферних опадів
2) Обґрунтувати вибір інструментів і технологій для реалізації системи прогнозування інтенсивності опадів

3) Розробити загальну архітектуру системи: модулі збору метеорологічних даних, попередньої обробки, навчання моделей, оцінювання та вебінтерфейсу
4) Розробити та реалізувати Transformer-регресор та XGBoost-класифікатор з гібридним механізмом узгодження прогнозів
5) Створити вебінтерфейс для візуалізації прогнозів, провести оцінювання якості моделей та продемонструвати роботу системи

5. Орієнтований перелік ілюстративного матеріалу діаграма використання, загальна архітектура моделі трансформера, графіки навчання Transformer-моделі та XGBoost-класифікатора, графіки оцінювання якості прогнозування, приклади роботи вебінтерфейсу системи прогнозування.

6. Дата видачі завдання 15.09.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Вибір теми роботи	07.10.25	
2.	Аналіз методів та засобів розв'язання задачі	20.04.26 – 26.04.26	
3.	Розробка архітектури та загальної структури системи	27.04.26 – 03.05.26	
4.	Розробка окремих підсистем	27.04.26 – 03.05.26	
5.	Програмна реалізація системи	04.05.26 – 10.05.26	
6.	Оформлення пояснювальної записки	11.05.25 – 22.05.26	
7.	Захист програмного забезпечення	14.05.26	
8.	Передзахист	04.06.26	
9.	Захист	17.06.26	

Студент _____ Дарина НАГАЇВСЬКА
(підпис) (прізвище та ініціали)

Керівник _____ Микита ГОЛОВАКІН
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота виконана на 86 сторінках, містить 25 рисунки, 4 таблиці, 2 додатки, 47 джерел у переліку посилань.

Мета роботи – розробка програмної системи прогнозування інтенсивності атмосферних опадів для семи міст України на основі гібридної архітектури регресійної моделі Transformer та XGBoost-класифікатора з горизонтом прогнозу 24 години.

Методи та засоби: Python 3.12, PyTorch 2.9.0, XGBoost 2.1, Pandas, NumPy, Scikit-learn, Streamlit, Plotly.

Основний зміст роботи: проведено аналіз методів прогнозування опадів; реалізовано пайплайн збору та обробки п'яти років погодинних спостережень для Києва, Харкова, Львова, Одеси, Дніпра, Чернігова та Ужгорода; навчено Transformer-регресор та XGBoost-класифікатор з механізмом раннього зупинення; реалізовано гібридний механізм узгодження прогнозів; розроблено вебінтерфейс з інтерактивною візуалізацією; проведено оцінювання за метриками RMSE, MAE, R^2, F1-Score, ROC-AUC.

Результат – програмна система прогнозування інтенсивності атмосферних опадів, що реалізує гібридну архітектуру регресійної моделі Transformer та XGBoost-класифікатора.

Ключові слова: машинне навчання, Transformer, XGBoost, прогнозування атмосферних опадів, часові ряди, глибоке навчання, гібридна архітектура, Streamlit, Python.

ANNOTATION

The thesis comprises 86 pages and includes 25 figures, 4 tables, 2 appendices and 47 references.

The aim of the thesis is to develop a software system for forecasting atmospheric precipitation intensity for seven cities of Ukraine based on a hybrid architecture combining a Transformer regression model and an XGBoost classifier with a 24-hour forecast horizon.

Methods and tools: Python 3.12, PyTorch 2.9.0, XGBoost 2.1, Pandas, NumPy, Scikit-learn, Streamlit, Plotly.

Main content of the work: an analysis of precipitation forecasting methods was conducted; a pipeline for collecting and processing five years of hourly observations for Kyiv, Kharkiv, Lviv, Odesa, Dnipro, Chernihiv and Uzhhorod was implemented; a Transformer regressor and an XGBoost classifier with an early stopping mechanism were trained; a hybrid forecast fusion mechanism was implemented; a web interface with interactive visualisation was developed; model evaluation was carried out using the RMSE, MAE, R^2, F1-Score and ROC-AUC metrics.

The result is a software system for forecasting atmospheric precipitation intensity implementing a hybrid architecture of a Transformer regression model and an XGBoost classifier.

Keywords: machine learning, Transformer, XGBoost, atmospheric precipitation forecasting, time series, deep learning, hybrid architecture, Streamlit, Python.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

API – Application Programming Interface –програмний інтерфейс застосунку.

CLS – Classification Token – спеціальний класифікаційний токен.

CSV – Comma-Separated Values – текстовий формат даних із роздільником комою.

ECMWF – European Centre for Medium-Range Weather Forecasts – Європейський центр середньострокових прогнозів погоди.

FFN – Feed-Forward Network – мережа прямого поширення у Transformer.

FPR – False Positive Rate – частка хибно позитивних результатів.

GELU – Gaussian Error Linear Unit – функція активації з гауссівською похибкою

GFS – Global Forecast System – глобальна система прогнозування погоди.

GPU – Graphics Processing Unit – графічний процесор.

GRU – Gated Recurrent Unit – рекурентний блок із вентилями.

JSON – JavaScript Object Notation – текстовий формат обміну даними.

LSTM – Long Short-Term Memory – довга короткострокова пам'ять.

MAE – Mean Absolute Error – середня абсолютна похибка.

MHA – Multi-Head Attention – механізм мультиголової уваги.

MLP – Multilayer Perceptron – багатошаровий перцептрон.

MSE – Mean Squared Error – середньоквадратична похибка.

NaN – Not a Number – відсутнє або некоректне числове значення.

NWP – Numerical Weather Prediction – чисельне прогнозування погоди.

PR-AUC – Area Under the Precision-Recall Curve – площа під кривою точність-повнота.

ReLU – Rectified Linear Unit – функція активації випрямленого лінійного блоку.

RMSE – Root Mean Square Error – корінь із середньоквадратичної похибки.

ROC – Receiver Operating Characteristic – робоча характеристика приймача.

ROC-AUC – Area Under the ROC Curve – площа під ROC-кривою.

TFT – Temporal Fusion Transformer – трансформер злиття часових ознак.

TPR – True Positive Rate – частка істинно позитивних результатів.

UTC – Coordinated Universal Time – всесвітній координований час.

XGBoost – eXtreme Gradient Boosting – екстремальний градієнтний бустинг.

ВСТУП

Прогнозування атмосферних опадів є однією з найважливіших задач сучасної метеорології. Точні прогнози кількості та інтенсивності опадів мають критичне значення для планування сільськогосподарської та будівельної діяльності, управління водними ресурсами, своєчасного попередження населення про небезпечні погодні явища та мінімізації збитків від стихійних лих.

Традиційні чисельні методи прогнозування погоди, які ґрунтуються на розв'язанні систем диференціальних рівнянь, вимагають значних обчислювальних ресурсів і складного обладнання. Водночас стрімкий розвиток методів машинного навчання та глибокого навчання відкриває нові можливості для ефективного прогнозування метеорологічних показників на основі історичних даних.

Особливий інтерес представляють архітектури на основі механізму самоуваги, зокрема трансформери, які довели свою ефективність у задачах аналізу часових рядів. У поєднанні з алгоритмами класифікації, такими як градієнтний бустинг, вони дозволяють побудувати гібридну систему, що здатна не лише оцінювати ймовірність випадіння опадів, але й регресійно передбачати їх погодинну інтенсивність на горизонті до 24 годин.

Актуальність теми.

Актуальність теми визначається зростаючою потребою у локалізованих прогнозах атмосферних опадів в умовах кліматичних змін, що супроводжуються збільшенням частоти та інтенсивності екстремальних погодних явищ. Більшість наявних рішень або є комерційними закритими системами, або орієнтованими на глобальний масштаб без адаптації до кліматичних особливостей конкретних регіонів.

Мета роботи.

Метою роботи є розробка програмної системи прогнозування інтенсивності атмосферних опадів для семи великих міст України із застосуванням гібридної архітектури, що поєднує Transformer-модель для регресійного прогнозування та

XGBoost-класифікатор для визначення факту наявності опадів, з горизонтом прогнозу 24 години.

Завдання роботи.

Для досягнення поставленої мети необхідно вирішити такі завдання. Насамперед здійснити аналіз сучасних методів та алгоритмів прогнозування атмосферних опадів та обґрунтувати доцільність використання гібридної архітектури. Дослідити програмні засоби для реалізації системи прогнозування інтенсивності атмосферних опадів та обґрунтувати їх вибір з урахуванням вимог до задачі. Розробити програмне забезпечення системи прогнозування інтенсивності атмосферних опадів, що включає модулі збору та попередньої обробки даних, навчання моделей машинного навчання, а також вебінтерфейс для візуалізації результатів прогнозування. Провести тестування розробленого програмного забезпечення, оцінити якість отриманих прогнозів та продемонструвати працездатність вебзастосунку.

Апробація результатів.

Результати роботи були представлені на студентській науковій конференції: XXIII міжнародна науково-практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики: безпека, стійкість, ІТ та екологічний моніторинг (до 40-річчя Чорнобильської катастрофи)» (доповідь «Application of the transformer architecture for atmospheric precipitation intensity forecasting», квітень 2026 р.). Результати апробації роботи наведено у додатку Б.

Структура й обсяг роботи. Дипломна робота складається зі вступу, 4 розділів, висновків, списку використаних джерел та додатків. Загальний обсяг роботи становить 86 сторінок, з яких сторінок основного 72 тексту. Робота містить 25 рисунків, таблиць. Список використаних джерел налічує 47 найменувань.

1 АНАЛІЗ МЕТОДІВ ПРОГНОЗУВАННЯ АТМОСФЕРНИХ ОПАДІВ

У даному розділі проведено аналіз предметної області задачі прогнозування атмосферних опадів та обґрунтовано вибір підходу до її розв'язання. Розглянуто постановку задачі, визначено вхідні та вихідні дані системи та потенційних користувачів. Проаналізовано традиційні чисельні методи прогнозування погоди та сучасні методи машинного навчання, зокрема рекурентні мережі, Transformer-архітектуру та алгоритми градієнтного бустингу. На основі проведеного аналізу обґрунтовано доцільність застосування гібридної архітектури що поєднує Transformer-модель для регресійного прогнозування та XGBoost-класифікатор для визначення факту наявності опадів.

1.1 Постановка задачі прогнозування опадів

Прогнозування атмосферних опадів є складною задачею аналізу часових рядів, оскільки метеорологічні процеси характеризуються високою нелінійністю, сезонністю та значною залежністю між параметрами атмосфери [1]. Для короткострокового прогнозування інтенсивності опадів особливо важливо враховувати зміну погодних умов у часі, оскільки інтенсивність опадів залежить від великої кількості факторів, таких як температура повітря, вологість, атмосферний тиск, характеристика вітру, хмарність та інші метеорологічні показники.

Особливістю задачі є необхідність одночасного розв’язання двох метеорологічних задач, а саме визначення наявності опадів методом класифікації та оцінку їх погодинної інтенсивності методом регресії. Поєднання цих підходів дозволяє отримати як якісний прогноз чи будуть опади чи ні, так і кількісний, скільки саме опадів очікується щогодини впродовж наступної доби для обраного

міста.

Система повинна приймати на вхід метеорологічні спостереження за останні 72 години. На виході система має формувати прогноз кількості опадів у мм/год та ймовірність у відсотках для обраного міста.

Для взаємодії з користувачем система повинна містити результати прогнозування інтенсивності опадів, а саме погодинні стовпчикові діаграми, теплову карту інтенсивності, кумулятивну криву накопичення опадів, текстовий аналітичний прогноз. Інтерфейс має бути простим та зрозумілим для користувача й забезпечувати наочне представлення результатів прогнозування.

Розроблена система може бути інтегрована в інформаційні системи підприємств, діяльність яких залежить від погодних умов. Потенційними користувачами системи можуть бути метеорологи, працівники аграрного сектору, служби моніторингу погодних умов, комунальні підприємства, а також звичайні користувачі, які потребують оперативного прогнозу опадів. Система може застосовуватись для планування сільськогосподарських та будівельних робіт, оцінювання ризику небезпечних погодних явищ, аналізу погодних умов та повсякденного використання.

1.2 Аналіз існуючих методів прогнозування атмосферних опадів

Прогнозування опадів є фундаментальним завданням метеорології, яке вирішується різними підходами від класичних чисельних методів до сучасних алгоритмів машинного та глибокого навчання. Огляд існуючих рішень дозволяє визначити переваги та обмеження кожного підходу і обґрунтувати вибір архітектурних рішень для розроблюваної системи.

1.2.1 Традиційні метеорологічні методи прогнозування

Традиційні методи прогнозування опадів базуються на чисельному моделюванні атмосферних процесів, а саме розв'язанні систем диференціальних рівнянь динаміки та термодинаміки. Найбільш поширеними є моделі NWP (Numerical Weather Prediction), серед яких виділяють глобальні та регіональні системи.

Моделі NWP базуються на складних математичних рівняннях, які враховують фізичні особливості погоди, такі як рух повітря, вологість і тиск, і є дорогими в експлуатації. Моделі NWP зазвичай генерують прогнози від декількох годин до тижнів. Поточний стан техніки для прогнозування опадів вимагає величезної обчислювальної потужності та ресурсів для запуску моделей чисельного прогнозування погоди в режимі реального часу [2].

Сучасні системи чисельного прогнозування погоди залишаються основним інструментом оперативної метеорології, однак їхній подальший розвиток обмежується низкою фундаментальних проблем. Зокрема, відзначається недостатня точність моделей при прогнозуванні рідкісних та екстремальних атмосферних явищ, а також обмежена здатність відтворювати складні багатомасштабні атмосферні процеси. Додаткові труднощі пов’язані з асиміляцією різнорідних даних у високороздільні моделі, що потребує значних обчислювальних ресурсів і складних процедур контролю якості.

Окремим викликом є стрімке зростання обсягів метеорологічних даних та обмежені можливості високопродуктивних обчислювальних систем, що стримує підвищення просторово-часової роздільної здатності моделей. Крім того, проблемними залишаються питання ансамблевого прогнозування, оперативного поширення результатів моделей та визначення пріоритетів розвитку NWP-систем [3].

Глобальна модель GFS (Global Forecast System) Американської метеорологічної служби NOAA забезпечує прогнози на горизонт до 16 діб із тимчасовим кроком 1-3 години та просторовою роздільністю 13 км. Схожу

архітектуру має Європейська модель ECMWF (European Centre for Medium-Range Weather Forecasts), яка традиційно вважається найточнішою у світі завдяки потужній системі асиміляції даних та покращеним параметризаціям фізичних процесів [4].

До основних переваг чисельних моделей NWP, зокрема GFS та ECMWF, належить створення довгострокових прогнозів. На відміну від локальних моделей, що охоплюють обмежені географічні зони, ECMWF забезпечує прогнозування погодних явищ у глобальному масштабі, а прогнози є загальнодоступними для користувачів [5].

Однак чисельні моделі мають також обмеження. Моделі NWP стикаються з обмеженнями у вигляді високої обчислювальної складності та недостатньої просторової роздільності для відтворення локальних подій. В результаті для запуску таких систем глобального масштабу необхідні суперкомп'ютери з десятками тисяч обчислювальних ядер та спеціалізованим програмним забезпеченням.

1.2.2 Методи машинного навчання для прогнозування часових рядів

Із розвитком машинного та глибокого навчання дослідники почали активно застосовувати нейронні мережі для прогнозування метеорологічних часових рядів. Першими отримали широке застосування рекурентні мережі – LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit).

Архітектура LSTM, запропонована Hochreiter та Schmidhuber у 1997 році, використовує спеціальні вентилі забування, введення та виведення для вибіркового збереження або забування інформації з попередніх часових кроків [6]. Це дозволяє мережі підтримувати довгострокову пам'ять і моделювати складні залежності у часових рядах. GRU є спрощеною версією LSTM з двома вентилями замість трьох, що зменшує кількість параметрів при збереженні більшості переваг.

Незважаючи на ефективність, рекурентні мережі мають серйозні обмеження

для довгих послідовностей. Обробка виконується послідовно, що виключає факт паралелізації при навчанні. В результаті градієнти вибухають через велику кількість кроків. Пам'ять обмежена і найдавніші взаємозалежності між даними забуваються моделею.

Архітектура Transformer, запропонована Vaswani et al. у 2017 році в роботі «Attention is All You Need», повністю відмовилась від рекурентних та згорткових шарів на користь механізму мультиголової самоуваги [7]. Кожен елемент вхідної послідовності отримує зважену суму всіх інших елементів, де ваги визначаються попарною схожістю запитів і ключів. Така архітектура дозволяє паралельно обробляти всю послідовність і прямо враховувати залежності між будь-якими елементами.

Для прогнозування часових рядів розроблено спеціалізовані варіанти трансформерів, а саме Temporal Fusion Transformer. Він поєднує механізм уваги з рекурентними шарами та змінними виборами ознак. Всі ці варіанти продемонстрували переваги над LSTM на бенчмарках прогнозування часових рядів [8].

Архітектура Transformer знайшла широке застосування у задачах метеорологічного прогнозування. Застосування Transformer-архітектури для прогнозів накопичених опадів за 24 години дозволило суттєво підвищити точність прогнозування порівняно з базовими моделями зокрема, показник загрози для опадів 50 мм покращився на 51,25% [9].

Алгоритми градієнтного бустингу, зокрема XGBoost, LightGBM та CatBoost, ефективно застосовуються для задач класифікації на табличних метеорологічних даних. XGBoost (eXtreme Gradient Boosting), запропонований Chen та Guestrin у 2016 році, будує ансамбль дерев рішень ітеративно: кожне нове дерево апроксимує негативний градієнт функції втрат відносно поточного прогнозу ансамблю [10]. Регуляризація L1 та L2, раннє зупинення та підтримка розріджених даних роблять XGBoost надзвичайно ефективним і стійким до перенавчання.

XGBoost демонструє перевагу у метеорологічній класифікації. Алгоритми на основі дерев рішень, зокрема XGBoost, є стійкими до шуму, ефективно працюють

у просторах ознак високої розмірності та забезпечують аналіз важливості ознак для інтерпретованості результатів, що особливо важливо для метеорологічних даних зі складними нелінійними залежностями [11].

Порівняльний аналіз розглянутих підходів свідчить про те, що рекурентні мережі поступаються Transformer-архітектурі за здатністю моделювати довгострокові залежності та можливістю паралельного навчання. Водночас для задачі бінарної класифікації наявності опадів на табличних метеорологічних даних алгоритми градієнтного бустингу, зокрема XGBoost, демонструють високу ефективність та стійкість до перенавчання. Поєднання цих двох підходів у гібридній архітектурі дозволяє розв'язати обидві підзадачі – регресію інтенсивності та класифікацію наявності опадів, з використанням методів, найбільш придатних для кожної з них.

1.3 Обґрунтування вибору гібридної архітектури

Аналіз розглянутих підходів до прогнозування атмосферних опадів свідчить про те, що жоден із методів окремо не здатен забезпечити оптимального розв'язання поставленої задачі. Традиційні чисельні моделі NWP потребують значних обчислювальних ресурсів і не адаптовані до локальних кліматичних умов.

Рекурентні мережі LSTM та GRU поступаються Transformer-архітектурі за здатністю моделювати довгострокові залежності у часових рядах через послідовний характер обробки. Водночас окреме застосування Transformer або XGBoost не дозволяє ефективно розв'язати одночасно дві підзадачі – класифікацію наявності опадів та регресійне прогнозування їх інтенсивності.

Метеорологічна задача прогнозування інтенсивності опадів складається з двох взаємопов'язаних підзадачі, а саме бінарної класифікації наявності опадів та регресійного прогнозування їх погодинної інтенсивності у мм/год на горизонті 24 години. Ці підзадачі мають різну природу і вимагають різних архітектурних підходів, що обґрунтовує доцільність гібридної архітектури.

Для задачі регресії інтенсивності опадів обрано Transformer-модель. Transformer зосереджується на навчанні складних часових залежностей, що робить його особливо придатним для прогнозування метеорологічних часових рядів із нелінійною динамікою [11]. Дослідження, що порівнювало чотири архітектури машинного навчання – XGBoost, MLP, Transformer та LSTM для прогнозування метеорологічних змінних у режимі реального часу, показало, що Transformer, навчений на повному наборі даних, досягав найнижчих значень похибки валідації у більшості розглянутих випадків [12].

Для задачі бінарної класифікації наявності опадів обрано XGBoost. XGBoost є ансамблевою технікою на основі градієнтного бустингу, де послідовність дерев рішень будується таким чином, що кожне дерево виправляє помилки попереднього, підвищуючи загальну точність моделі [13]. Алгоритми на основі бустингу є домінуючими у задачах із табличними структурованими даними, оскільки вони ітеративно зменшують залишкові помилки і при належному налаштуванні часто досягають вищої дискримінаційної здатності порівняно з іншими методами.

Доцільність поєднання цих двох підходів підтверджується у літературі. Гібридна архітектура Transformer і XGBoost дозволяє Transformer навчати складні часові залежності, тоді як XGBoost підвищує стійкість фінальних прогнозів, зменшуючи перенавчання та покращуючи узагальнення [11]. Поєднання прогностичної потужності Transformer із інтерпретованістю та ефективністю XGBoost забезпечує більш точне прогнозування у метеорологічних застосуваннях реального часу.

У розроблюваній системі моделі навчаються незалежно одна від одної. Об'єднання результатів відбувається лише на етапі інференсу. XGBoost-класифікатор визначає факт наявності опадів на основі 72-годинного вікна спостережень, після чого Transformer формує регресійний прогноз інтенсивності на 24 години. Такий підхід забезпечує модульність системи. Кожну із моделей можна вдосконалити незалежно, що підвищує гнучкість і масштабованість розробленої системи прогнозування інтенсивності опадів.

2 МЕТОДИ ТА ЗАСОБИ РЕАЛІЗАЦІЇ СИСТЕМИ ПРОГНОЗУВАННЯ ІНТЕНСИВНОСТІ АТМОСФЕРНИХ ОПАДІВ

У даному розділі описані методи, алгоритми та програмні засоби що використовуються при реалізації системи прогнозування інтенсивності атмосферних опадів. Розглянуто теоретичні основи архітектури Transformer для аналізу часових рядів та архітектури XGBoost для бінарної класифікації.

Окрему увагу приділено методам попередньої обробки та нормалізації метеорологічних даних, формуванню вхідних ознак та лагових характеристик, а також гібридному механізму узгодження прогнозів двох незалежно навчених моделей на етапі інференсу.

Також у розділі описано методи оцінювання якості прогнозування для регресійної та класифікаційної моделей, а також наведено огляд мови програмування та програмних засобів що використовуються для реалізації системи.

2.1 Архітектура Transformer-моделі для аналізу часових рядів

У сучасних дослідженнях прогнозування метеорологічних показників усе більше уваги приділяється методам машинного навчання та глибинного навчання, здатним враховувати складні нелінійні взаємозв’язки та тривалі часові залежності.

Саме тому в наукових роботах останніх років широко досліджуються алгоритми машинного навчання, що дозволяють підвищити точність короткострокових прогнозів.

Трансформери як нейромережева архітектура демонструють суттєві переваги у задачах, що передбачають аналіз кількох взаємопов’язаних часових рядів, зокрема при прогнозуванні погодних умов [17]. У цьому контексті часто

аналізуються параметри зі складними взаємозв'язками, такими як температура, вологість та тиск. Застосування трансформерів дозволяє фіксувати ці взаємозалежності, тим самим покращуючи точність прогнозування [7]. Завдяки механізму self-attention трансформери фіксують довгострокові залежності між даними, що особливо важливо при прогнозуванні інтенсивності опадів у короткі проміжки часу. Використання цих моделей у поєднанні з оптимізованими алгоритмами навчання, такими як Adam, та функціями втрат, зокрема MSE (Mean Squared Error) або MAE (Mean Absolute Error), дозволяє досягти точності прогнозування, недосяжної для класичних моделей машинного навчання [7].

Для прогнозування інтенсивності атмосферних опадів рекомендується застосовувати модель на основі архітектури трансформер, яка вперше запропонована у роботі «Attention Is All You Need». Трансформер – це нейронна мережа для обробки послідовностей, яка повністю базується на механізмі уваги та не використовує рекурентні або згорткові шари [7]. Основою трансформерів є механізм уваги, який дозволяє моделі визначати, які елементи вхідної послідовності є найбільш важливими для формування прогнозу. Це дає змогу ефективно враховувати довгострокові залежності між даними, що є особливо важливим у задачах прогнозування погоди.

Трансформери як нейромережева архітектура мають модульну структуру типу «encoder–decoder», що складається з двох основних компонентів: кодувальника та декодера.

Кодувальник приймає вхідну послідовність і перетворює її у внутрішнє числове представлення. Він складається з шести однакових шарів. Кожен шар містить два основні підшари, а саме механізм багатоголовкової самоуваги та позиційну повнозв’язну мережу прямого поширення.

Декодер також складається зі стеку з шести однакових шарів. На відміну від кодувальника, кожен шар декодера містить три підшари: маскований механізм багатоголовкової самоуваги, механізм багатоголовкової уваги до виходу кодувальника, позиційну повнозв’язну мережу прямого поширення. Як і в кодувальнику, навколо кожного підшару використовуються залишкові з’єднання

та нормалізація шару. Загальну архітектуру моделі трансформера зображено на рисунку 2.1.

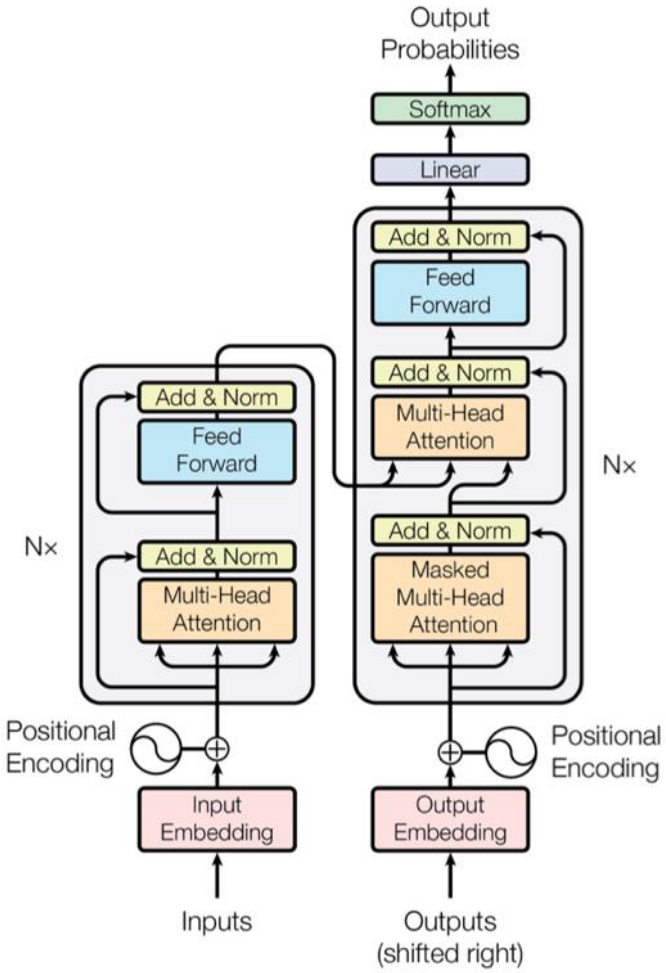

Рисунок 2.1 – Загальна архітектура моделі Transformer

Оскільки Transformer не має рекурентної структури, він не може природно відстежувати порядок елементів у послідовності. Для вирішення цієї проблеми застосовується позиційне кодування. Трансформерні моделі є інваріантними до перестановок за своєю природою та не мають природного розуміння порядку послідовності, тому позиційне кодування додається до вхідних ембедингів перед подачею до енкодера для збереження інформації про часовий порядок елементів. Позиційне кодування використовує синусоїдальні функції різних частот, що дозволяє моделі розрізняти елементи послідовності за їх відносною позицією у

часі.

Ключовим елементом архітектури є механізм самоуваги. Для кожного елемента вхідної послідовності формуються три вектори: запит (Query, Q), ключ (Key, K) та значення (Value, V).

$$\mathrm{Q} = X \cdot W^{Q}$$
$$K = X \cdot W^{K}$$
$$V = X \cdot W^{V} \quad (1)$$

де X ∈ $\mathbb{R}^{n \times d}$ d – розмір вкладення,

$W^{Q} \in \mathbb{R}^{n \times d_Q}$, $W^{K} \in \mathbb{R}^{n \times d_K}$, $W^{V} \in \mathbb{R}^{n \times d_V}$ – визначені матриці ваг, що навчаються

Вектори запиту, ключа та значення формуються відповідно до формули (1).

Оцінка уваги між елементами обчислюється як скалярний добуток запитів і ключів, нормований на квадратний корінь із розмірності ключів, після чого застосовується функція SoftMax.

$$Attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d_Q}}) \cdot V \quad (2)$$

де Q (запити) – те, що ми шукаємо, поточний фокус уваги,

K (ключі) – мітки, які показують, яку інформацію містить кожен елемент,

V (значення) – сама інформація, яку ми отримаємо, якщо знайдемо збіг

Обчислення ваг уваги здійснюється відповідно до формули (2).

Мультиголова увага (multi-head attention) паралельно виконує механізм уваги h разів із різними проєкційними матрицями, що дозволяє моделі одночасно фокусуватися на різних аспектах вхідної послідовності.

$$head = Attention(XW_i^Q, XW_i^K, XW_i^V)$$
$$MultiHead(X) = Concat(head_1, \dots\ head_2)W^O \quad (3)$$

де $W_i^Q \in \mathbb{R}^{n \times d_Q}$ – матриця проєкції запитів,

$W_i^K \in \mathbb{R}^{n \times d_K}$ – матриця проєкції ключів,

$W_i^V \in \mathbb{R}^{n \times d_V}$ – матриця проєкції значень,

$W_i^V \in \mathbb{R}^{hd_V \times n}$ – вихідна матриця проєкції

Механізм мультиголової уваги, наведений у формулі (3), дозволяє моделі

одночасно аналізувати різні типи залежностей у часовому ряді.

Для задачі прогнозування інтенсивності атмосферних опадів модель отримує на вхід послідовність метеорологічних показників і формує на виході регресійний прогноз на майбутній період. Стандартний механізм самоуваги Transformer має квадратичну часову складність відносно довжини вхідної послідовності, проте при фіксованій довжині вхідного вікна це обмеження не є критичним для практичного застосування. Навчання здійснюється з учителем із використанням регресійних функцій втрат MSE або MAE та оптимізатора Adam із зворотним поширенням помилки.

Таким чином, ключовою перевагою трансформера є здатність до паралельної обробки всіх елементів вхідної послідовності. На відміну від рекурентних мереж, де обробка є послідовною, трансформер обчислює ваги уваги між всіма парами елементів одночасно. Це усуває проблему зникаючих та вибухових градієнтів при роботі з довгими послідовностями та дозволяє ефективно використовувати паралелізм сучасних GPU при навчанні.

2.2 Архітектура XGBoost-моделі для бінарної класифікації

XGBoost (eXtreme Gradient Boosting) – це алгоритм машинного навчання на основі ансамблю дерев рішень із градієнтним бустингом, запропонований Chen та Guestrin у 2016 році. XGBoost є потужним алгоритмом машинного навчання, що використовує дерева рішень із градієнтним бустингом і відомий своєю високою точністю, швидкістю та ефективністю, що робить його популярним вибором для різноманітних задач прогнозування, включаючи класифікацію та регресію [18].

Принцип роботи алгоритму полягає в ітеративній побудові ансамблю дерев рішень. Модель навчається адитивним способом, а саме на кожній ітерації будується нове дерево, яке апроксимує залишкові помилки поточного ансамблю, а фінальний прогноз для кожного прикладу є сумою прогнозів усіх дерев ансамблю.

Цільова функція XGBoost складається з двох компонентів – функції втрат та

члена регуляризації (4):

$$\text{obj}(\theta) = \sum_{i}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (4)$$

де $l(y_i, \hat{y}_i)$ – функція втрат, яка обчислює різницю між істинним значенням y_i та прогнозованим $\hat{y}_i$,

$\Omega(f_k)$ – регуляризація, яка запобігає появі надмірно складних дерев

Ключовою відмінністю XGBoost від звичайного градієнтного бустингу є використання апроксимації другого порядку Тейлора для оптимізації цільової функції. XGBoost оптимізує апроксимацію функції втрат другого порядку Тейлора з двома параметрами регуляризації, що дозволяє отримати аналітичне рішення для оптимальних ваг листків дерева та прискорює навчання [19].

XGBoost використовує кілька додаткових механізмів для запобігання перенавчанню. Параметри subsample, max_leaves та max_depth застосовуються для зменшення перенавчання та покращення моделі, а коефіцієнт швидкості навчання контролює вагу нових дерев, що додаються до моделі, знижуючи швидкість адаптації моделі до тренувальних даних. Завдяки перевагам обчислювальної ефективності, аналізу важливості ознак та обробці пропущених значень XGBoost широко застосовується у задачах регресії та класифікації на різноманітних типах даних.

Для задачі бінарної класифікації наявності атмосферних опадів XGBoost використовує логістичну функцію втрат. На виході модель формує ймовірність належності спостереження до класу «опади є» через сигмоїдну функцію. Бінарне рішення отримується шляхом застосування порогового значення до вихідної ймовірності. Серед помітних переваг XGBoost для класифікаційних задач виділяють обробку пропущених значень, зменшення перенавчання та підвищення обчислювальної ефективності завдяки паралельним та розподіленим обчисленням.

Підсумовуючи, можна виділити, що архітектура XGBoost є обґрунтованим вибором для задачі бінарної класифікації наявності атмосферних опадів. Використання апроксимації другого порядку Тейлора забезпечує точнішу оптимізацію цільової функції порівняно з методами першого порядку. Здатність

обробляти пропущені значення та формувати оцінку важливості ознак робить XGBoost особливо придатним для роботи з реальними метеорологічними часовими рядами, де можливі пропуски у спостереженнях. Вихідна ймовірність логістичної функції втрат безпосередньо інтерпретується як ймовірність випадіння опадів, що є інформативним результатом для кінцевого користувача системи прогнозування.

2.3 Попередня обробка та нормалізація метеорологічних даних

Попередня обробка даних є обов'язковим етапом підготовки метеорологічних часових рядів до навчання моделей машинного навчання. Якість вхідних даних безпосередньо впливає на точність прогнозування, оскільки наявність пропущених значень, викидів та різномасштабних ознак може суттєво знизити ефективність навчання моделі.

Метеорологічні спостереження є типовими часовими рядами, що характеризуються наявністю пропущених значень внаслідок відмов датчиків, перебоїв у передачі даних або технічного обслуговування обладнання.

Методи імпутації пропущених значень у часових рядах включають заповнення середнім значенням, стохастичну регресійну імпутацію та методи на основі авторегресійного моделювання. Точність імпутації залежить від відповідності між представницькою моделлю часового ряду в алгоритмі та реальною структурою даних. Для метеорологічних часових рядів з регулярним кроком у часі ефективним методом є лінійна інтерполяція між сусідніми спостереженнями, оскільки метеорологічні показники змінюються неперервно і короткочасні пропуски можна відновити з достатньою точністю.

Перевірка фізично допустимих меж ознак є важливим кроком контролю якості метеорологічних даних. Для кожного метеорологічного показника існують фізично обґрунтовані діапазони допустимих значень. Наприклад, відносна вологість повітря не може перевищувати 100% або бути від'ємною, а атмосферний тиск знаходиться в межах фізично можливих значень для приземного шару

атмосфери. Значення, що виходять за межі допустимих діапазонів, замінюються на пропущені з подальшою імпутацією.

Метеорологічні ознаки мають різні одиниці вимірювання та масштаби значень, що може негативно впливати на навчання моделей, чутливих до масштабу вхідних даних.

Стандартизація перетворює кожну ознаку до нульового середнього та одиничного стандартного відхилення:

$$z = \frac{x - \mu}{\sigma} \tag{5}$$

де x – значення ознаки,

μ – середнє значення ознаки на тренувальній вибірці,

σ – стандартне відхилення

Мінімакс-нормалізація масштабує значення до діапазону [0, 1]:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

де x – значення ознаки, яке потрібно масштабувати,

x_{min} – мінімальне значення серед усіх даних цієї ознаки,

x_{max} – максимальне значення серед усіх даних цієї ознаки

Важливо, що параметри нормалізації обчислюються виключно на тренувальній вибірці та застосовуються до валідаційної і тестової вибірок. Параметри функції нормалізації мінімум і максимум для мінімакс-масштабування, середнє та стандартне відхилення для стандартизації повинні обчислюватися лише на даних, доступних під час навчання, без урахування даних поза тренувальною вибіркою.

Для часових рядів розбиття даних на тренувальну, валідаційну та тестову вибірки має виконуватися з обов'язковим урахуванням часової впорядкованості. Застосування стандартної крос-валідації до часових рядів призводить до того, що тренувальні вибірки містять дані з майбутнього відносно тестової вибірки. Для коректної оцінки здатності моделі до прогнозування в умовах часової динаміки, що змінюється, широко застосовується часова крос-валідація, яка використовує

концепцію ковзних вікон та створює часове розділення між тренувальними та оціночними даними для запобігання витоку інформації.

У висновку можна сказати, що коректна попередня обробка метеорологічних даних із дотриманням часової впорядкованості при розбитті вибірок є необхідною умовою отримання достовірних оцінок якості прогнозування та запобігання завищенню показників точності моделі.

2.4 Формування вхідних ознак та лагових характеристик

Проєктування ознак є критичним етапом побудови систем прогнозування часових рядів, оскільки якість та інформативність вхідних даних безпосередньо визначають здатність моделі виявляти закономірності та формувати точні прогнози. Сирі метеорологічні спостереження потребують перетворення у форму, що дозволяє моделі ефективно відтворювати часові залежності, сезонні закономірності та просторовий контекст.

Лагові ознаки є одним із найбільш поширених підходів до представлення часових залежностей у задачах прогнозування. Авторегресійний підхід є одним із найпоширеніших методів розв'язання задач прогнозування часових рядів. Спостереження моделюються за допомогою множинної регресії з використанням попередніх лагів як предикторних змінних.

Для задачі прогнозування атмосферних опадів лагові значення цільової змінної інтенсивності опадів за попередні 1, 2, 3, 6 та 12 годин дозволяють моделі враховувати короткострокову та середньострокову динаміку розвитку опадових подій. Включення лагових регресорів різної глибини суттєво покращує кореляцію між вхідними даними та цільовою змінною, дозволяючи відтворювати як короткострокові, так і довгострокові часові закономірності.

Для формування вхідної послідовності застосовується метод ковзного вікна фіксованої довжини. Кожен навчальний приклад формується як послідовність метеорологічних спостережень за попередні T годин, де T – розмір вікна. Підхід на основі ковзного вікна дозволяє структурувати часові ряди у форму, придатну для навчання моделей машинного навчання, забезпечуючи фіксований розмір вхідного тензора при збереженні часової впорядкованості спостережень. Вікно розміром 72 години забезпечує модель достатнім контекстом для відтворення добових та міждобових закономірностей розвитку опадових систем.

Таким чином, формування вхідних ознак та лагових характеристик є невід'ємною складовою побудови ефективної системи прогнозування атмосферних опадів. Використання лагових значень цільової змінної дозволяє моделі враховувати короткострокову та середньострокову динаміку розвитку опадових подій. Підхід на основі ковзного вікна структурує часові ряди у форму придатну для навчання моделей із збереженням часової впорядкованості. Включення географічних координат дозволяє єдиній моделі враховувати кліматичні відмінності між містами різних регіонів без необхідності навчання окремих моделей для кожного населеного пункту.

2.5 Гібридний механізм узгодження прогнозів

Задача прогнозування атмосферних опадів має змішану природу: вона одночасно включає дискретну підзадачу визначення факту наявності опадів та неперервну підзадачу оцінювання їх інтенсивності. Такі задачі природно розв'язуються за допомогою двоетапних гібридних архітектур, де перший етап відповідає за класифікацію, а другий за регресію. Двоетапний підхід, що поєднує класифікацію та регресію, дозволяє отримати кращі прогнози порівняно з одноетапними регресійними моделями, оскільки явне моделювання факту наявності події підвищує точність оцінювання її кількісних характеристик.

Каскадний механізм узгодження передбачає послідовне застосування двох моделей. Класифікатор визначає факт настання події та повертає бінарне рішення або ймовірність, після чого регресійна модель формує кількісний прогноз. Фінальний результат отримується шляхом застосування індикаторної функції класифікатора до виходу регресійної моделі, якщо класифікатор визначає відсутність події, регресійний прогноз обнуляється. Двоетапна архітектура дозволяє врахувати дискретно-неперервну природу розподілу опадів, що є сумішшю точкової маси в нулі та неперервного розподілу для ненульових значень інтенсивності.

Важливою особливістю каскадних гібридних систем є можливість незалежного навчання компонентів. Гібридні моделі здатні подолати обмеження окремих алгоритмів завдяки поєднанню їх переваг, при цьому кожен компонент може бути замінений або вдосконалений незалежно, що підвищує гнучкість та масштабованість системи. Об'єднання результатів відбувається виключно на етапі інференсу, що усуває ризик взаємного перенавчання компонентів та дозволяє оптимізувати гіперпараметри кожної моделі незалежно.

Таким чином, гібридний механізм узгодження прогнозів, що поєднує XGBoost-класифікатор та Transformer-регресор у каскадній архітектурі, є обґрунтованим рішенням для задачі прогнозування атмосферних опадів. Класифікатор усуває характерну для регресійних моделей проблему хибних ненульових прогнозів у суху погоду, тоді як Transformer забезпечує детальний погодинний прогноз інтенсивності при підтвердженій наявності опадів. Незалежне навчання компонентів дозволяє оптимізувати кожну модель окремо та за необхідності замінювати їх без перенавчання системи в цілому.

2.6 Оцінювання якості прогнозування інтенсивності атмосферних опадів

Оцінювання якості прогнозування є важливим етапом розробки систем машинного навчання, що дозволяє об'єктивно порівнювати моделі між собою та з базовими підходами. Оскільки розроблена система розв'язує дві підзадачі – регресію інтенсивності опадів та бінарну класифікацію їх наявності для кожної підзадачі використовується окремий набір метрик. Для об'єктивного оцінювання результатів модель порівнюється з тривіальним базовим підходом, що завжди передбачає нульову інтенсивність опадів, що є природним базовим рівнем для задач із нерівномірним розподілом цільової змінної.

2.6.1 Метрики якості регресійної моделі

Регресійна модель Transformer розв'язує задачу кількісного прогнозування погодинної інтенсивності атмосферних опадів у мм/год на горизонті 24 години. Оскільки цільова змінна має нерівномірний розподіл із переважанням нульових та малих значень і рідкісними екстремальними подіями, для комплексної оцінки якості прогнозування використовується набір взаємодоповнюючих метрик. Використання кількох метрик у поєднанні з візуалізацією результатів є необхідною умовою для надійної та інтерпретованої оцінки якості регресійних моделей прогнозування, оскільки окремі метрики можуть давати неповну або хибну картину ефективності моделі.

Для оцінювання якості прогнозування часових рядів недостатньо лише вимірювати величину похибок за допомогою метрик MAE або MSE, важливо також оцінювати як ці похибки змінюються в часі та наскільки точно модель відтворює сезонні та циклічні закономірності. Метрика R^2, що пояснює варіативність усього набору даних, є корисною для аналізу довгострокових тенденцій у часових рядах,

тоді як RMSE є чутливою до великих помилок і дозволяє оцінити здатність моделі справлятися зі значними відхиленнями.

Для оцінювання якості Transformer-моделі, що прогнозує погодинну інтенсивність опадів, використовуються такі метрики: RMSE, MAE, MSE та коефіцієнт детермінації R^2.

Середня абсолютна похибка (MAE) обчислює середнє абсолютне відхилення між прогнозованими та фактичними значеннями:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_{\hat{i}}| \tag{7}$$

де n – кількість спостережень або точок даних,

i – індекс конкретної точки даних,

y_i – фактичне (істинне) значення для точки даних,

$y_{\hat{i}}$ – прогнозоване значення для точки даних,

$|y_i - y_{\hat{i}}|$ – абсолютна похибка

MAE є стійкою до викидів метрикою, оскільки всі відхилення враховуються з однаковою вагою незалежно від їх величини. MAE рекомендується використовувати коли всі похибки мають однакову важливість, тоді як RMSE є більш доцільною коли великі відхилення є особливо небажаними, оскільки вона штрафує їх непропорційно більше.

Середньоквадратична похибка (MSE) обчислює середнє значення квадратів відхилень:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_{\hat{i}})^2 \tag{8}$$

де n – кількість спостережень або точок даних,

i – індекс конкретної точки даних,

y_i – фактичне (істинне) значення для точки даних,

$y_{\hat{i}}$ – прогнозоване значення для точки даних,

$(y_i - y_{\hat{i}})$ – абсолютна похибка

Квадратична природа MSE робить її чутливою до великих відхилень, значні помилки штрафуються непропорційно більше, що є важливим для задач де екстремальні похибки є критичними.

Корінь із середньоквадратичної похибки (RMSE) є квадратним коренем із MSE:

$$RMSE = \sqrt{MSE} \tag{9}$$

де MSE – середньоквадратична похибка

Коефіцієнт детермінації R^2 показує яку частку дисперсії цільової змінної пояснює модель:

$$R^2 = 1 - \frac{\sum(y_i - y_{\hat{i}})^2}{\sum(y_i - \bar{y})^2} \tag{10}$$

де y_i – фактичне (істинне) значення для точки даних,

$\bar{y}$ – середнє значення цільової змінної,

$y_{\hat{i}}$ – прогнозоване значення для точки даних

Значення $R^2 = 1$ відповідає ідеальному прогнозу, $R^2 = 0$ означає що модель не краща за передбачення середнього значення, від'ємні значення свідчать про те що модель гірша за тривіальний базовий підхід. Від'ємні значення R^2 є характерними для задач прогнозування опадів із нерівномірним розподілом даних, де переважна більшість спостережень належить до класу нульових або малих значень.

При інтерпретації регресійних метрик важливо враховувати їх шкали та відносний характер оцінювання. MAE та RMSE є масштабозалежними метриками. Їх значення виражаються в тих самих одиницях що й цільова змінна (мм/год), тому їх абсолютні значення інтерпретуються відносно типового діапазону цільової змінної. Порівняння моделей лише за абсолютними значеннями MAE або RMSE без урахування простих базових підходів є поширеною помилкою при оцінюванні прогнозних моделей. Результати завжди слід інтерпретувати відносно базової моделі.

Для MSE інтерпретація аналогічна MAE та RMSE, проте значення є квадратом одиниць цільової змінної, що ускладнює його безпосередню

інтерпретацію, тому MSE використовується переважно як функція втрат під час навчання, тоді як MAE та RMSE є більш зручними для звітування результатів.

R^2 є чутливим до викидів і може давати завищену оцінку точності за наявності екстремальних значень, тому його слід використовувати разом з MAE та RMSE для забезпечення повної картини якості моделі.

Отже у висновку, для комплексного оцінювання якості регресійної моделі прогнозування інтенсивності атмосферних опадів доцільно використовувати набір взаємодоповнюючих метрик, а саме MAE, MSE, RMSE та R^2. Кожна з метрик характеризує окремий аспект точності прогнозування. MAE відображає середню величину похибки, MSE та RMSE є чутливими до великих відхилень, а коефіцієнт детермінації R^2 оцінює здатність моделі пояснювати варіативність цільової змінної. Спільне використання цих метрик дозволяє отримати об’єктивну та інтерпретовану оцінку ефективності Transformer-моделі при прогнозуванні атмосферних опадів.

2.6.2 Метрики якості класифікаційної моделі

Для оцінювання якості XGBoost-класифікатора, що визначає факт наявності атмосферних опадів, використовується набір метрик на основі матриці помилок. Матриця помилок є основою для обчислення більшості класифікаційних метрик і містить чотири складові: TP (істинно позитивні) – випадки опадів правильно визначені як опади, TN (істинно негативні) – відсутність опадів правильно визначена як відсутність, FP (хибно позитивні) – відсутність опадів хибно визначена як опади, FN (хибно негативні) – опади хибно визначені як відсутність. Усі основні метрики класифікації це Accuracy, Precision, Recall та F1-Score і обчислюються вони на основі цих чотирьох складових матриці помилок.

Accuracy (точність) визначає частку правильно класифікованих спостережень серед усіх:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

де TP (True Positive) – істинно позитивний результат,

TN (True Negative) – істинно негативний результат,

FP (False Positive) – хибно позитивний результат,

FN (False Negative) – хибно негативний результат

При роботі з незбалансованими даними Accuracy може бути оманливою метрикою, оскільки модель що завжди передбачає більш поширений клас може отримати високу точність без реальної здатності до розрізнення класів. Саме тому для задачі класифікації наявності опадів, де класи можуть бути незбалансованими, Accuracy розглядається лише як допоміжна метрика.

Precision визначає частку правильно визначених позитивних випадків серед усіх передбачених позитивних, тобто наскільки можна довіряти прогнозу моделі коли вона передбачає наявність опадів:

$$Precision = \frac{TP}{(TP + FP)} \quad (12)$$

де TP (True Positive) – істинно позитивний результат,

FP (False Positive) – хибно позитивний результат

Recall визначає частку правильно виявлених позитивних випадків серед усіх фактично позитивних, тобто яку частку реальних опадових подій модель здатна виявити:

$$Recall = \frac{TP}{(TP + FN)} \quad (13)$$

де TP (True Positive) – істинно позитивний результат,

FN (False Negative) – хибно негативний результат

Precision показує наскільки можна довіряти моделі коли вона передбачає позитивний клас, тоді як Recall показує яку частку фактично позитивних прикладів модель здатна виявити. Ці дві метрики знаходяться у взаємній залежності, підвищення однієї зазвичай призводить до зниження іншої. Для задачі прогнозування опадів висока повнота є пріоритетною, оскільки пропуск реальної опадової події є більш критичною помилкою ніж хибне спрацювання.

F1-Score є гармонічним середнім між Precision та Recall:

$$F1 = 2 \cdot \frac{(Precision \cdot Recall)}{(Precision + Recall)} \quad (14)$$

де Precision та Recall визначаються відповідно до формул (12) – (13).

F1-Score досягає найвищого значення 1 при ідеальних Precision та Recall і найнижчого значення 0 при повній відсутності корисних передбачень. Гармонічне середнє є більш інформативним ніж арифметичне, оскільки воно штрафує моделі з великим дисбалансом між Precision та Recall. F1-Score є рекомендованою метрикою при незбалансованих класах, оскільки арифметична точність може бути оманливою через домінування більш поширеного класу.

Для комплексної оцінки здатності моделі розділяти класи при різних порогових значеннях використовуються порогово-незалежні метрики ROC-AUC та PR-AUC. ROC-AUC оцінює ймовірність того що випадково вибраний позитивний приклад отримає вищу оцінку класифікатора ніж випадково вибраний негативний, і візуалізується як крива залежності при різних порогових значеннях. Значення ROC-AUC = 0.5 відповідає випадковому класифікатору, значення 1.0 – ідеальному розділенню класів. PR-крива є більш чутливою до незбалансованості класів порівняно з ROC-кривою, що робить PR-AUC більш інформативною метрикою для задач де позитивний клас зустрічається значно рідше, оскільки ROC-AUC може давати надмірно оптимістичну оцінку в таких умовах.

Таким чином можна підсумувати, що для оцінювання якості класифікаційної моделі прогнозування наявності атмосферних опадів доцільно використовувати комплекс взаємодоповнюючих метрик – Accuracy, Precision, Recall, F1-Score, ROC-AUC та PR-AUC. Кожна з метрик характеризує окремий аспект роботи класифікатора, а їх спільне використання дозволяє об’єктивно оцінити здатність моделі розрізняти класи в умовах незбалансованих даних.

2.7 Мова програмування та засоби розробки

Підрозділ описує мову програмування та програмний стек що використовуються для реалізації системи прогнозування атмосферних опадів. Вибір інструментів здійснювався з урахуванням вимог до продуктивності, зручності розробки та наявності спеціалізованих бібліотек для задач машинного навчання.

2.7.1 Мова програмування Python

Мовою реалізації системи обрано Python версії 3.12, яка є стандартом у галузі наукових обчислень та машинного навчання. Основні переваги Python для даної задачі це розвинена екосистема бібліотек для обробки даних та машинного навчання, висока читабельність коду, активна спільнота та велика кількість актуальних досліджень з відкритим кодом [40]. Python став основною мовою для розробки застосунків у сферах науки про дані та машинного навчання, дозволяючи дослідникам швидко реалізовувати алгоритми навіть без глибокого досвіду системного програмування. Додатковою перевагою є відкритість мови та велика кількість актуальних досліджень із відкритим кодом, що спрощує відтворення та верифікацію результатів

2.7.2 Бібліотеки обробки даних

Для роботи з числовими масивами та статистичними операціями використовуються бібліотеки NumPy версії 1.26.4 та Pandas версії 2.3.3. Бібліотека NumPy надає ефективні операції над багатовимірними масивами. Pandas забезпечує зручний інтерфейс для завантаження, фільтрації та трансформації таблиць даних, що є критично важливим при роботі з великими наборами метеорологічних

спостережень.

Для попередньої обробки вхідних ознак використовується бібліотека Scikit-learn, зокрема компонент StandardScaler який реалізує стандартизацію даних до нульового середнього та одиничного стандартного відхилення. Scikit-learn також надає інструменти для обчислення метрик якості класифікації та регресії. Для серіалізації навчених об'єктів масштабування використовується стандартна бібліотека pickle, для збереження результатів експериментів – бібліотека json.

2.7.3 Фреймворк для навчання нейронних мереж

Для навчання нейронних мереж обрано фреймворк PyTorch версії 2.9.0. PyTorch став популярним інструментом у дослідницькій спільноті глибокого навчання завдяки поєднанню орієнтованості на зручність використання з ретельним урахуванням продуктивності.

Ключовою відмінністю PyTorch є динамічний граф обчислень та підхід у стилі Python, що контрастує з підходом TensorFlow на основі статичного графа. PyTorch домінує у наукових публікаціях зокрема у 2023 році він використовувався приблизно у 80% статей конференції NeurIPS що вказували фреймворк. Динамічний граф дозволяє змінювати структуру мережі під час виконання, що є особливо зручним при експериментуванні з нестандартними архітектурами, зокрема Transformer із CLS-токеном для регресії.

2.7.4 Бібліотека градієнтного бустингу

Для реалізації XGBoost-класифікатора використовується бібліотека xgboost версії 2.1. Бібліотека надає ефективну реалізацію алгоритму градієнтного бустингу з підтримкою паралельного навчання на CPU, вбудованими механізмами регуляризації та інтерфейсом сумісним із Scikit-learn. Підтримка раннього

зупинення та вбудованої крос-валідації дозволяє ефективно підбирати гіперпараметри моделі. Бібліотека також надає інструменти аналізу важливості ознак, що є корисним для інтерпретації результатів класифікації метеорологічних подій.

2.7.5 Засоби розробки вебінтерфейсу та візуалізації

Для розробки вебінтерфейсу системи прогнозування використовується фреймворк Streamlit. Streamlit дозволяє створювати інтерактивні вебзастосунки безпосередньо на мові Python, що суттєво скорочує час розробки інтерфейсу. Фреймворк забезпечує автоматичне оновлення інтерфейсу при зміні вхідних параметрів та підтримує інтеграцію з популярними бібліотеками візуалізації.

Для побудови інтерактивних графіків у вебінтерфейсі використовується бібліотека Plotly, яка надає широкий набір типів діаграм із підтримкою інтерактивності – масштабування, фільтрації та відображення деталей при наведенні курсору.

Для збереження графіків навчання моделей використовується бібліотека Matplotlib, що є стандартним інструментом візуалізації в екосистемі Python.

Можливість легко писати та підтримувати код Python у поєднанні з великою кількістю спеціалізованих фреймворків та бібліотек, таких як Matplotlib та Pandas, забезпечує високу продуктивність розробки наукових застосунків.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПРОГНОЗУВАННЯ ІНТЕНСИВНОСТІ АТМОСФЕРНИХ ОПАДІВ

Розділ присвячено програмній реалізації системи прогнозування атмосферних опадів. У ньому розглянуто загальну архітектуру програмної системи, реалізацію модулів збору та попередньої обробки метеорологічних даних, побудову Transformer-моделі для прогнозування інтенсивності опадів і XGBoost-класифікатора для визначення факту їх наявності. Також описано реалізацію вебінтерфейсу засобами Streamlit для взаємодії користувача з системою.

3.1 Архітектура програмної системи прогнозування інтенсивності атмосферних опадів

Програмна система прогнозування атмосферних опадів побудована за модульним принципом, що забезпечує функціональну незалежність окремих компонентів та спрощує їх подальше вдосконалення або заміну. Кожен модуль виконує чітко визначену функцію та взаємодіє з іншими через збережені артефакти – файли даних, масштабувальники та ваги моделей, без прямої залежності між вихідним кодом компонентів.

Система складається з п’яти основних функціональних модулів: data_collection, data_preprocessing, models, evaluation та interface. Кожен компонент функціонально незалежний і взаємодіє з іншими через збережені артефакти, нормалізовані набори даних, масштабувальники та збережені ваги моделей.

Модуль data_collection відповідає за завантаження погодинних метеорологічних спостережень через Weather API для семи міст України. Модуль data_preprocessing виконує очищення, нормалізацію та розбиття даних на вибірки, зберігаючи окремі масштабувальники для кожного міста. Модуль models містить

скрипти навчання Transformer-моделі та XGBoost-класифікатора. Модуль evaluation реалізує обчислення метрик якості та побудову графіків оцінювання. Модуль interface реалізує Streamlit-застосунок для взаємодії з користувачем. Взаємодія між модулями здійснюється виключно через збережені артефакти без прямих викликів між компонентами.

Варіанти використання системи та взаємодія між користувачем і функціональними модулями наведені на рисунку 3.1.

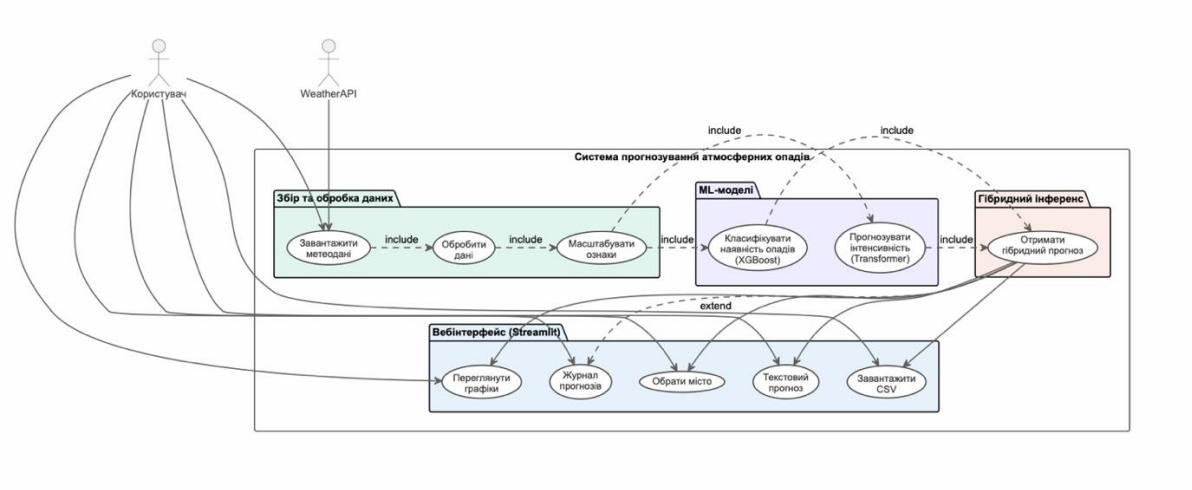

Рисунок 3.1 – Діаграма варіантів використання системи прогнозування інтенсивності атмосферних опадів

Саме такий підхід забезпечує можливість незалежного запуску кожного модуля, що спрощує налагодження та повторне навчання моделей без необхідності перезапуску всього пайплайну.

3.2 Реалізація модуля збору метеорологічних даних

Збір метеорологічних даних здійснюється через REST API сервісу WeatherAPI, який надає безкоштовний доступ до погодинних історичних спостережень. Взаємодія з API реалізована за допомогою бібліотеки requests. Запит

формується з параметрами міста, дати та ключа доступу і у відповідь отримується у форматі JSON та розбирається у табличну структуру засобами бібліотеки Pandas.

Дані завантажуються для семи міст України – Києва, Харкова, Львова, Одеси, Дніпра, Чернігова та Ужгорода за часовий діапазон з березня 2021 року по березень 2026 року, що становить п'ять років погодинних спостережень. Вибір міст забезпечує географічне охоплення різних кліматичних зон України, від помірно-континентального клімату центру та сходу до карпатського клімату заходу та морського впливу на півдні.

Набір із 26 метеорологічних ознак включає такі показники: температура повітря та точки роси, відносна вологість, атмосферний тиск, швидкість та напрямок вітру, пориви вітру, зональна та меридіональна складові вектора вітру, хмарність, горизонтальна видимість, сніговий покрив, код погодних умов та індикатор денного часу. Додатково генеруються циклічні часові ознаки sin та cos від години доби й місяця року, що дозволяє моделі враховувати добові та сезонні цикли атмосферних процесів без розривів на межах циклів. Результатом роботи модуля є CSV-файли із сирими погодинними спостереженнями окремо для кожного міста, які передаються на вхід модуля попередньої обробки даних.

У висновку, реалізований модуль збору метеорологічних даних забезпечує автоматизоване отримання погодинних спостережень із зовнішнього REST API та формування структурованих наборів даних для подальшого аналізу. Використання широкого набору метеорологічних і часових ознак, а також охоплення різних кліматичних регіонів України створює інформативну основу для навчання моделей прогнозування атмосферних опадів.

3.3 Реалізація модуля попередньої обробки даних

Модуль попередньої обробки даних реалізує послідовний пайплайн перетворень вхідних метеорологічних спостережень для підготовки їх до навчання моделей машинного навчання. Попередня обробка даних є критичним етапом будь-

якого пайплайну машинного навчання і може суттєво впливати як на продуктивність так і на ефективність навчання, оскільки реальні часові ряди часто характеризуються нерегулярністю, скошеністю розподілу та наявністю викидів, що може призводити до різкого погіршення якості моделі. Пайплайн складається з чотирьох послідовних етапів: очищення даних, формування додаткових ознак, розбиття на вибірки та нормалізація.

На етапі очищення для кожної метеорологічної ознаки перевіряються фізично допустимі межі значень. Зокрема, температура повітря обмежується діапазоном від - 50°C до +50°C, відносна вологість – від 0% до 100%, атмосферний тиск – від 870 до 1085 гПа, швидкість вітру – від 0 до 100 м/с. Значення що виходять за встановлені межі замінюються на NaN. Далі дані приводяться до регулярної погодинної сітки в часовому поясі UTC, а пропущені значення заповнюються методом лінійної інтерполяції між сусідніми спостереженнями. При пропусках понад 6 послідовних годин відповідний діапазон повністю виключається з навчальної вибірки. Такий підхід є обґрунтованим для метеорологічних часових рядів оскільки погодні показники змінюються неперервно і короткочасні пропуски можна відновити з достатньою точністю.

Для покращення прогнозування формуються lag-ознаки на основі попередніх значень цільової змінної precip_mm за 1, 2, 3, 6 та 12 годин. Ці ознаки дозволяють моделі враховувати короткострокову та середньострокову динаміку розвитку опадових подій безпосередньо через вхідний вектор ознак. Додатково формуються циклічні часові ознаки через тригонометричні перетворення години доби та місяця року що забезпечує коректне представлення циклічної природи часових показників без розривів на межах циклів.

Дані розподіляються на тренувальну, валідаційну та тестову вибірки з обов'язковим урахуванням часової впорядкованості. Тестова вибірка містить лише спостереження що хронологічно йдуть після тренувальної та валідаційної, що виключає витік інформації з майбутніх спостережень до навчальної вибірки та забезпечує коректну оцінку здатності моделі до узагальнення на нових даних. Розбиття на вибірки виконується суворо за часом без перемішування: перші 70%

хронологічно впорядкованих даних утворюють тренувальну вибірку, наступні 15% – валідаційну, останні 15% – тестову.

Нормалізація виконується окремо для кожного з семи міст. Для вхідних метеорологічних ознак застосовується StandardScaler з бібліотеки Scikit-learn, параметри якого середнє μ та стандартне відхилення σ обчислюються виключно на тренувальній вибірці та застосовуються до валідаційної і тестової без повторного навчання.

Стандартизація за z-оцінкою генерує найточніші прогнози порівняно з ненормалізованими даними. Покращення точності становить близько 49% за метрикою RMSE та 56% за MAE, що підтверджує критичну важливість нормалізації для навчання моделей на часових рядах [46]. Для вхідних метеорологічних ознак застосовується StandardScaler який масштабує значення до нульового середнього та одиничного стандартного відхилення.

Цільова змінна precip_mm попередньо піддається логарифмічному перетворенню за формулою:

$$x' = \log(1 + x)$$

де x – вихідне значення ознаки,

x'– значення ознаки після логарифмічного перетворення

Логарифмічне перетворення змінює розподіл ознак і може використовуватись для отримання наближеного до нормального розподілу для ознак з іншими розподілами, що є особливо важливим коли прогностична модель передбачає нормальний розподіл даних. Після логарифмічного перетворення до цільової змінної також застосовується StandardScaler. Зворотне перетворення під час інференсу виконується як:

$$x = \exp(x') - 1x$$

де x'– значення ознаки після логарифмічного перетворення,

x – відновлене значення ознаки у вихідному масштабі

Для кожного міста зберігаються два окремі об'єкти масштабування: feature_scaler для вхідних ознак та target_scaler для цільової змінної у форматі pkl засобами бібліотеки pickle. Ці об'єкти завантажуються під час інференсу у

вебінтерфейсі для коректного перетворення вхідних даних та зворотного масштабування прогнозів у фізичні одиниці мм/год.

Таким чином, реалізований модуль попередньої обробки даних забезпечує повний цикл підготовки метеорологічних часових рядів до навчання моделей машинного навчання, включаючи очищення даних, формування ознак, часовий поділ вибірок та нормалізацію.

3.4 Реалізація Transformer-моделі

Для задачі прогнозування опадів у даній роботі використовується архітектура Encoder-only Transformer з CLS-токеном. Вибір обмеженого енкодером варіанта обґрунтований структурою задачі, адже необхідно сформувати весь прогноз одночасно на основі глобального представлення вхідної послідовності. Архітектура моделі складається з трьох послідовних логічних блоків.

Перший блок це вхідна обробка, де проектує 26-вимірний вектор метеорологічних ознак у 128-вимірний простір моделі (d_model = 128) через лінійний шар з нормалізацією LayerNorm та активацією ReLU. До отриманої послідовності додаються позиційні кодування та спеціальний CLS-токен.

Другий блок – стек шарів енкодера, який складається з 4 ідентичних шарів TransformerEncoderLayer. Кожен шар містить підшар мультиголової самоуваги з 8 головами та підшар прямого поширення (FFN) з прихованою розмірністю 512. Використовується архітектура Pre-LN (Pre-Layer Normalization), де нормалізація виконується перед обчисленнями кожного підшару, а не після. Це забезпечує стабільніше навчання глибоких трансформерів [9].

Третій блок – регресійна голівка, яка реалізована як тришаровий MLP (Multi-Layer Perceptron) з активаціями GELU між шарами. Вона приймає на вхід вихідний стан CLS-токену (вектор розмірності 128) та формує прогноз погодинної інтенсивності опадів на 24 години. Активація GELU обрана замість ReLU через кращу поведінку при малих і від'ємних значеннях у нормалізованому просторі

ознак.

Навчання моделі виконується з використанням оптимізатора AdamW із L2-регуляризацією. Функцією втрат є комбінована регресійна втрата MSE та MAE, яка поєднує середньоквадратичну похибку, що штрафує великі відхилення, та середньоабсолютну похибку, яка є стійкішою до нульових спостережень. Розклад навчання включає лінійне попереднє нагрівання протягом перших 5 епох і подальше зниження швидкості навчання за допомогою ReduceLROnPlateau. Завдяки Early Stopping навчання зупинилось на 22-й епосі з найкращим val_loss = 0.4665. Динаміку навчання наведено на рисунку 3.2.

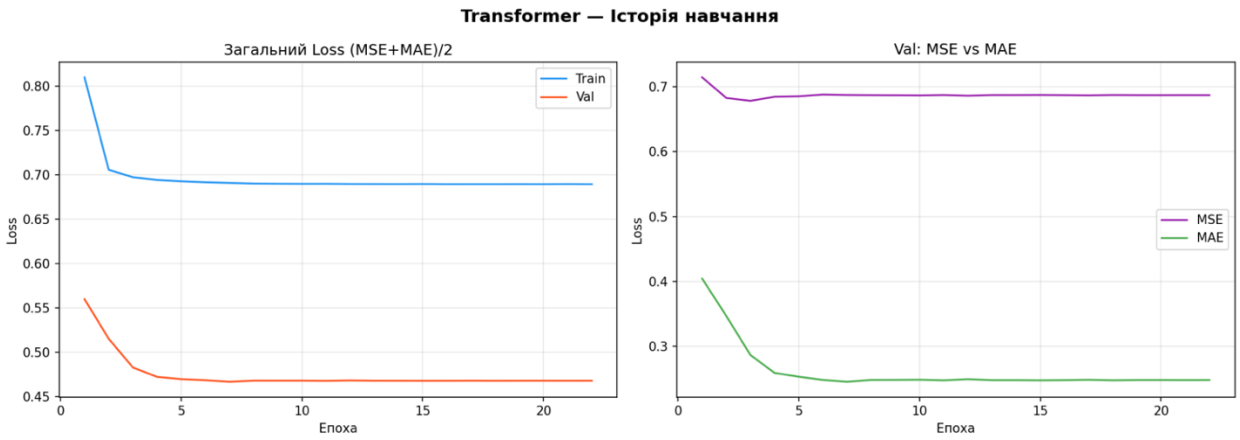

Рисунок 3.2 – Історія навчання Transformer-моделі, загальний loss та MSE і MAE на валідаційній вибірці

З графіку видно, що модель швидко сходиться протягом перших 5 епох і далі стабілізується без ознак перенавчання, train і val loss залишаються близькими.

Можемо зробити висновок, що реалізована Transformer-модель на основі Encoder-only архітектури з CLS-токеном забезпечує ефективне вилучення глобальних залежностей у метеорологічних часових рядах та формування узагальненого представлення для подальшого прогнозування. Використання багатошарової енкодерної структури, сучасних механізмів нормалізації та регуляризації, а також комбінованої функції втрат дозволило досягти стабільного навчання без ознак перенавчання. Отримані результати підтверджують придатність

обраної архітектури для задачі прогнозування погодинної інтенсивності атмосферних опадів.

3.5 Реалізація XGBoost-класифікатора

Навчання XGBoost-класифікатора виконується на збалансованому датасеті, де цільова змінна приймає значення 1, якщо максимальна інтенсивність опадів у наступні 24 години перевищує поріг 0.1 мм/год, і 0 в іншому випадку. Такий поріг обрано для відфільтрування шуму вимірювань та вологості повітря, яка може проявлятися у вигляді надмалих значень.

Вектор ознак для класифікатора формується статистичною агрегацією нормалізованого вхідного вікна з 72 годин. Для кожної з 26 метеорологічних ознак обчислюються середнє, стандартне відхилення, мінімум та максимум за всім вікном, значення останнього часового кроку, а також ковзні середні за вікнами 3, 6 і 12 годин. Таким чином розмірність вхідного вектора класифікатора становить 26 × 8 = 208 ознак. Перевірка приналежності вікна одному місту виконується явно, вікна на межах між містами виключаються з навчання.

Модель навчається з такими гіперпараметрами: n_estimators=300, max_depth=6, learning_rate=0.05, subsample=0.8, colsample_bytree=0.8, reg_alpha=0.1, reg_lambda=1.0. Незбалансованість класів враховується через параметр scale_pos_weight, що обчислюється як відношення кількості негативних прикладів до позитивних. Застосовується Early Stopping і навчання зупинилось на ітерації 70. Динаміку навчання наведено на рисунку 3.3.

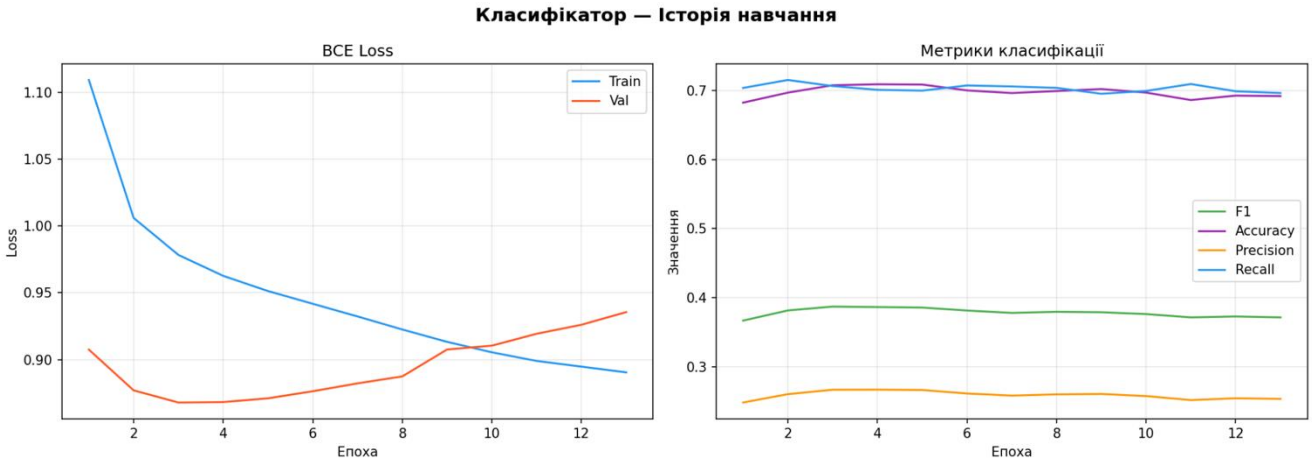

Рисунок 3.3 – Історія навчання XGBoost-класифікатора

Таким чином, реалізований XGBoost-класифікатор забезпечує ефективне розв’язання задачі бінарного визначення наявності атмосферних опадів на основі статистично агрегованих ознак із часових вікон. Використання узгодженого набору гіперпараметрів, механізму балансування класів та ранньої зупинки дозволило досягти стабільного навчання та зменшити ризик перенавчання. Отримана модель є придатною для інтеграції в гібридну систему прогнозування опадів.

3.6 Реалізація вебінтерфейсу засобами Streamlit

Для розробки інтерфейсу користувача обрано фреймворк Streamlit, що дозволяє швидко створювати інтерактивні веб-застосунки на Python. Бібліотека Plotly використовується для побудови інтерактивних графіків із підтримкою масштабування, підказок і анімації [47].

Вебінтерфейс реалізовано як односторінковий інтерактивний застосунок, який містить елементи керування для вибору міста, перегляду вхідних метеорологічних даних та відображення результатів прогнозування.

Інтерфейс забезпечує інтеграцію з попередньо навченими моделями Transformer та XGBoost, які завантажуються при старті застосунку.

Ключовою перевагою Streamlit для даної системи є механізм кешування обчислень через декоратори. Завдяки цьому важкі артефакти такі як ваги нейронної мережі, масштабувальники завантажуються лише один раз при першому запуску та зберігаються в пам'яті між сесіями. Перемикання між містами у інтерфейсі не призводить до повторного завантаження моделей, що забезпечує швидку відповідь застосунку.

Таким чином, реалізований вебінтерфейс на основі Streamlit забезпечує зручну взаємодію користувача з системою прогнозування атмосферних опадів. Інтеграція інтерактивних елементів керування та візуалізацій дозволяє оперативно отримувати та аналізувати результати роботи моделей Transformer та XGBoost. Використання механізму кешування та оптимізованої архітектури застосунку забезпечує високу швидкодію та ефективність роботи вебсистеми.

4 ТЕСТУВАННЯ ТА ДЕМОНСТРАЦІЯ ПРОГРАМНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ІНТЕНСИВНОСТІ ОПАДІВ

У даному розділі наведено результати тестування та практичної демонстрації розробленої програмної системи прогнозування інтенсивності атмосферних опадів. Розглянуто вимоги до апаратного та програмного забезпечення, необхідні для коректного запуску та стабільної роботи системи, а також описано процес інсталяції та налаштування програмного середовища.

Окрему увагу приділено демонстрації функціональних можливостей вебзастосунку, зокрема взаємодії користувача з інтерфейсом, процесу отримання вхідних даних, формування прогнозів за допомогою моделей машинного навчання та візуалізації результатів.

Також у розділі наведено результати оцінювання якості роботи регресійної та класифікаційної моделей на тестових даних. Проведено аналіз точності прогнозування інтенсивності опадів та ефективності визначення факту їх наявності з використанням відповідних метрик якості.

4.1 Вимоги до технічного забезпечення та інсталяція

Програмна система прогнозування інтенсивності атмосферних опадів розроблена як локальний вебзастосунок та не потребує серверної інфраструктури для розгортання. Для коректної роботи системи необхідне виконання мінімальних вимог до апаратного та програмного забезпечення.

Мінімальна конфігурація включає процесор з тактовою частотою не менше 2.0 ГГц та кількістю ядер не менше 4, оперативну пам'ять обсягом не менше 8 ГБ, вільний простір на диску не менше 5 ГБ для зберігання датасету, навчених моделей та скейлерів. Наявність відеокарти (GPU) не є обов'язковою – усі обчислення

виконуються на центральному процесорі (CPU). Для завантаження метеорологічних даних через Weather API необхідне підключення до мережі Інтернет.

Система сумісна з операційними системами Windows 10/11, macOS 12 та вище, Ubuntu 20.04 та вище. Обов'язковою умовою є наявність встановленого інтерпретатора Python версії 3.10 або вище та пакетного менеджера pip. Рекомендується використання менеджера середовищ Anaconda, оскільки система розроблялась та тестувалась у середовищі Anaconda.

Перелік основних залежностей із зазначенням версій наведено у таблиці 4.1.

Таблиця 4.1 – Основні залежності програмної системи

Бібліотека	Версія	Призначення
Python	3.12	Мова програмування
streamlit	≥ 1.35	Вебінтерфейс
torch (PyTorch)	2.9.0	Transformer-модель
xgboost	≥ 2.0	Класифікатор
scikit-learn	≥ 1.4	Масштабування даних
pandas	2.3.3	Обробка табличних даних
numpy	1.26.4	Числові обчислення
plotly	≥ 5.20	Інтерактивні графіки
requests	≥ 2.31	Запити до Weather API

Інсталяція системи виконується у чотири послідовні кроки. На першому кроці виконується завантаження репозиторію проекту на локальний комп'ютер та перехід до кореневої директорії проекту. На другому кроці створюється та активується ізольоване віртуальне середовище Anaconda з інтерпретатором Python версії 3.12, що забезпечує незалежність залежностей системи від інших встановлених пакетів. На третьому кроці у активованому середовищі

встановлюються всі необхідні бібліотеки відповідно до файлу requirements.txt, що містить перелік залежностей із зафіксованими версіями. На четвертому кроці виконується безпосередній запуск вебзастосунку командою streamlit run app.py.

Після виконання четвертого кроку Streamlit автоматично запускає локальний вебсервер. Усі чотири команди виконуються послідовно у терміналі операційної системи.

Після виконання останньої команди застосунок автоматично відкривається у браузері. При першому запуску система завантажує в пам'ять ваги Transformer-моделей та XGBoost-класифікатори для всіх семи міст, що займає 8-12 секунд. Подальші перемикання між містами виконуються протягом 1-2 секунд завдяки механізму кешування.

Перед отриманням першого прогнозу необхідно виконати три підготовчі кроки через бічну панель застосунку послідовно натиснути «Завантажити нові дані» для отримання метеорологічних спостережень через Weather API. Після успішного завантаження натиснути «Обробити дані» для очищення та нормалізації, потім «Масштабування» для побудови скейлерів. Після завершення всіх трьох кроків натиснути «Оновити прогноз» і система готова до роботи.

На основі наведеного опису можна зробити висновок, що для коректного функціонування програмної системи, визначено мінімальні вимоги до апаратного та програмного забезпечення, які забезпечують стабільне виконання всіх етапів обробки даних та роботи моделей машинного навчання. Описаний процес інсталяції дозволяє відтворити середовище виконання системи шляхом послідовного встановлення залежностей, створення ізольованого середовища та запуску вебзастосунку.

Завдяки використанню Streamlit та механізмів кешування забезпечується швидкий старт системи та оперативна робота інтерфейсу після первинного завантаження моделей, що підвищує зручність використання програмного продукту.

4.2 Демонстрація функціоналу вебзастосунку прогнозування

Інтерфейс користувача реалізований у вигляді вебзастосунку Streamlit і запускається командою streamlit run app.py. Застосунок відображається у браузері і містить бічну панель для налаштувань та основну робочу область.

Він є кінцевою точкою взаємодії користувача з розробленою системою машинного навчання і об'єднує в собі всі компоненти, а саме завантаження даних, обробку, інференс двох моделей і відображення результатів. Загальний вигляд інтерфейсу наведено на рисунку 4.1.

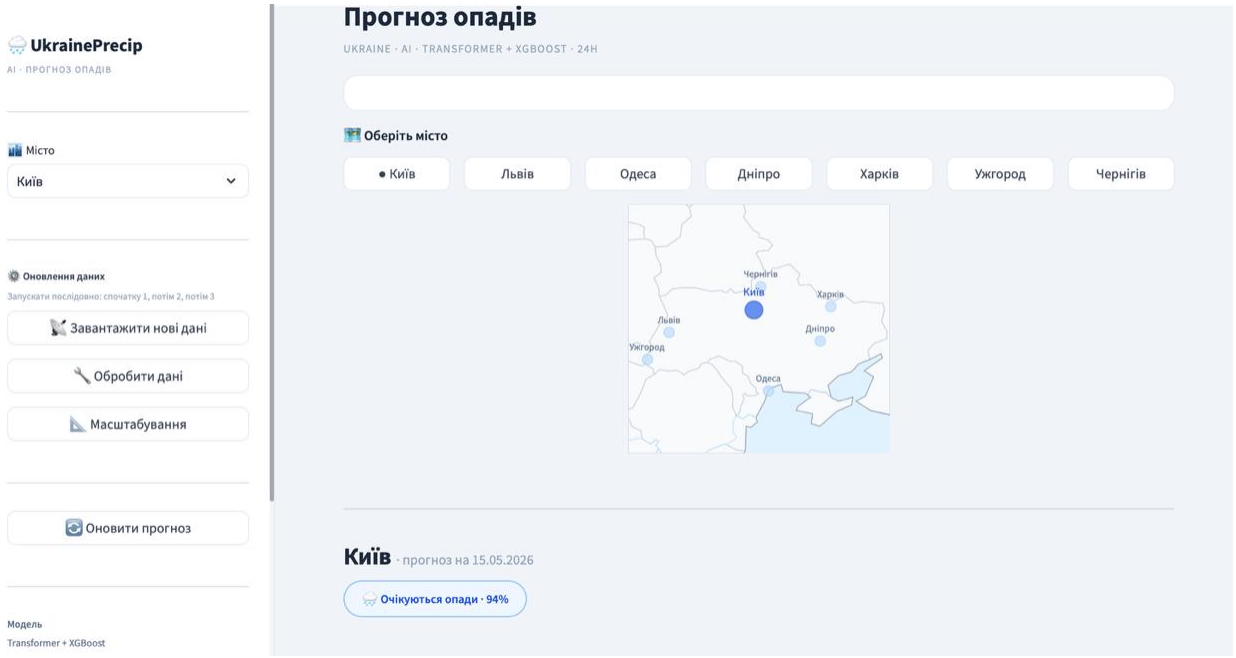

Рисунок 4.1 – Загальний вигляд інтерфейсу користувача

Бічна панель містить випадаючий список для вибору міста, кнопки завантаження нових даних, обробка даних та масштабування. Бічна панель застосунку зображена на рисунку 4.2.

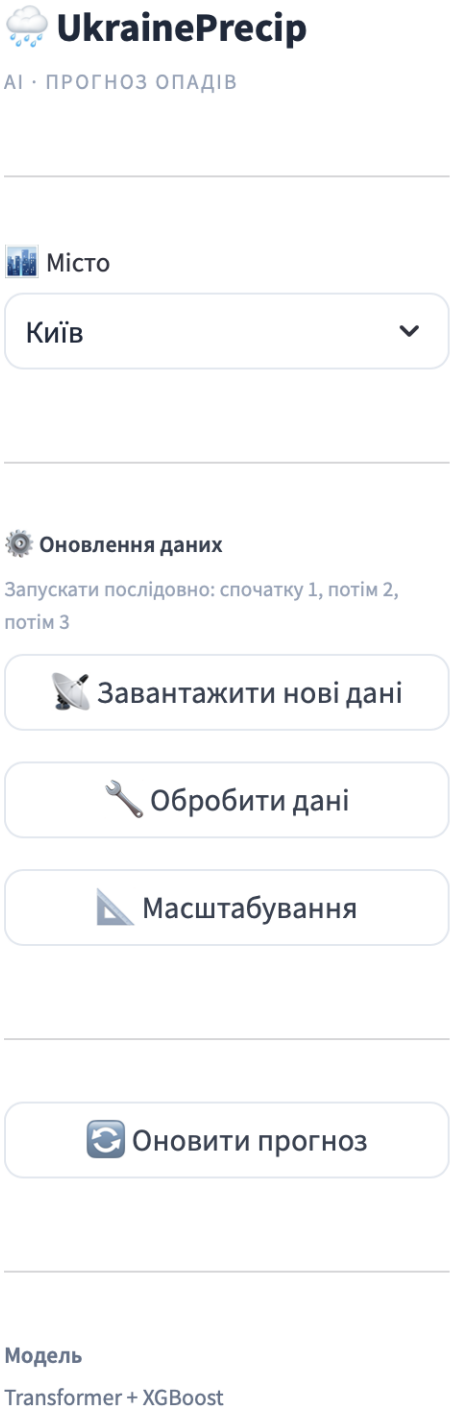

Рисунок 4.2 – Бічна панель застосунку

Випадаючий список дозволяє обрати одне з семи міст: Київ, Львів, Одеса, Дніпро, Харків, Ужгород, Чернігів. При зміні міста автоматично перераховується прогноз, завантажуються відповідні scalers для цього міста і береться вікно

спостережень саме по ньому. Випадаючий список з вибором міст зображений на рисунку 4.3.

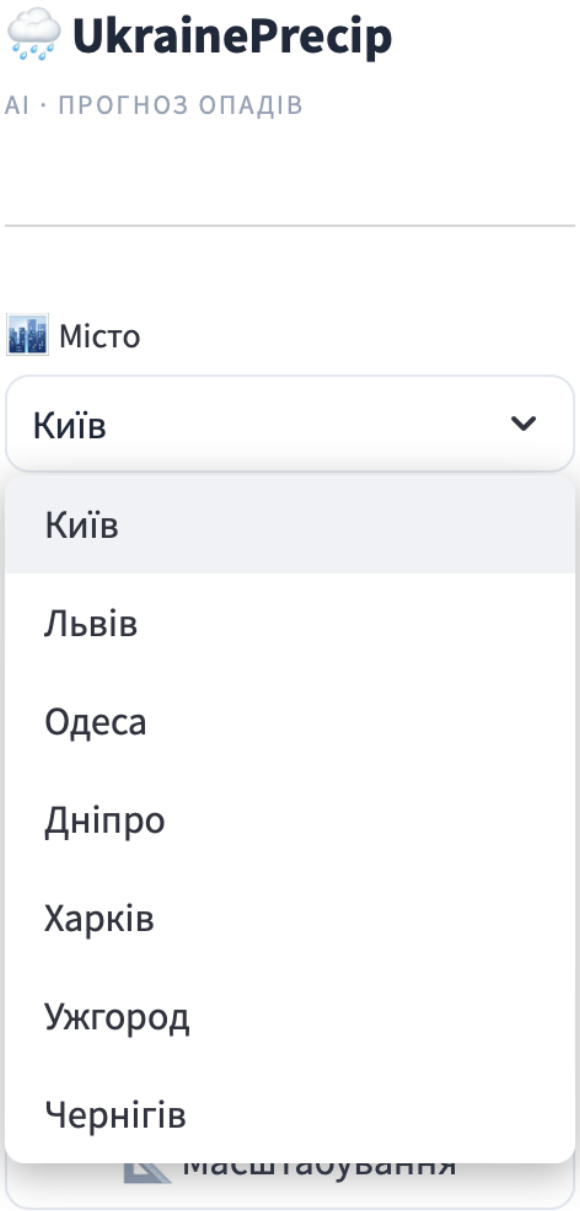

Рисунок 4.3 – Вибір міста з випадного списку

У блоці оновлення даних розташовані три кнопки, які необхідно запускати послідовно. Вони відтворюють у інтерфейсі той самий процес, який раніше виконувався вручну через термінал. Блок оновлення даних зображений на рисунку 4.4.

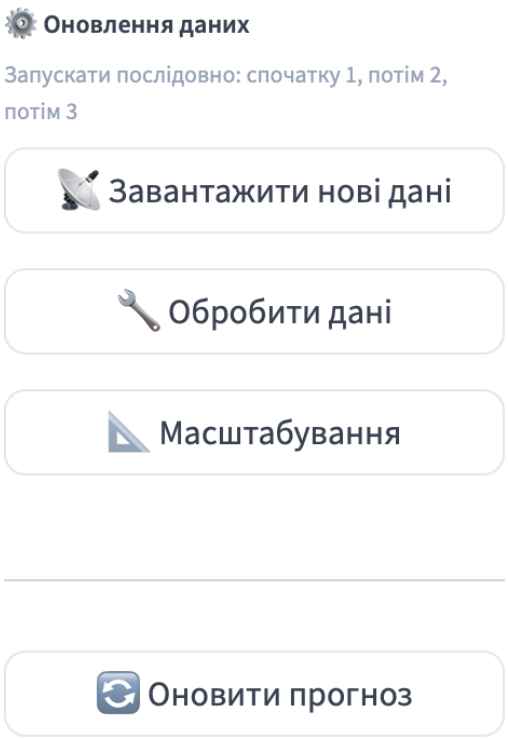

Рисунок 4.4 – Блок оновлення даних

Перша кнопка – «Завантажити нові дані» запускає скрипт update_data.py, який звертається до WeatherAPI і оновлює сирі CSV-файли з погодинними спостереженнями для всіх семи міст. Після виконання застосунок показує зелене повідомлення із переліком міст, для яких дані оновлено.

Друга кнопка – «Обробити дані» запускає preprocess_weather_data.py з двома аргументами: шляхом до сирих даних і шляхом до папки з обробленими даними. Скрипт виконує п'ять кроків – перевірку сентинельних значень, перевірку часової неперервності, заповнення пропусків, розбиття на вибірки і нормалізацію з збереженням scalers.

Третя кнопка – «Перебудувати scalers» запускає rebuild_feature_scalers.py, який перезаписує feature scalers з урахуванням усіх 26 ознак. Цей крок необхідний для забезпечення сумісності між scaler і моделлю.

Кнопка «Оновити прогноз» очищає кеш і примусово перераховує прогноз з нових даних. Вона потрібна тому що Streamlit кешує результати на 5 хвилин і без неї старий прогноз залишатиметься на екрані навіть після оновлення даних.

У нижній частині панелі відображається довідкова інформація, а саме назва моделі, горизонт прогнозу 24 години і поріг опадів 0.1 мм/год. Довідкова інформація з бічної панелі наведена на рисунку 4.5.

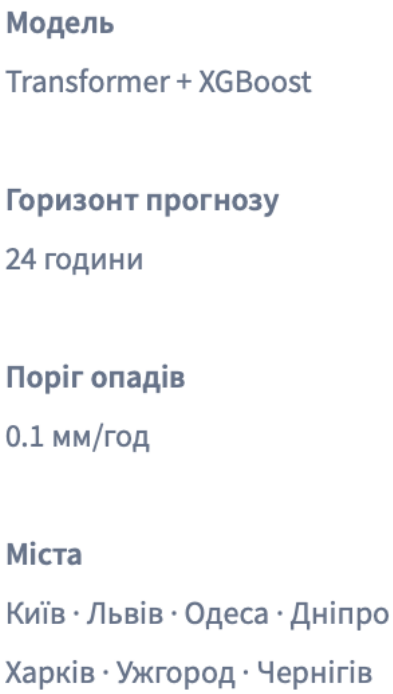

Рисунок 4.5 – Довідкова інформація

Основна область складається з інтерактивної карти вибору міста, чотирьох метричних карток та чотирьох вкладок: «Графіки», «Текстовий прогноз», «Таблиця даних» та «Журнал прогнозів». Основна робоча область користувача наведена на рисунку 4.6.

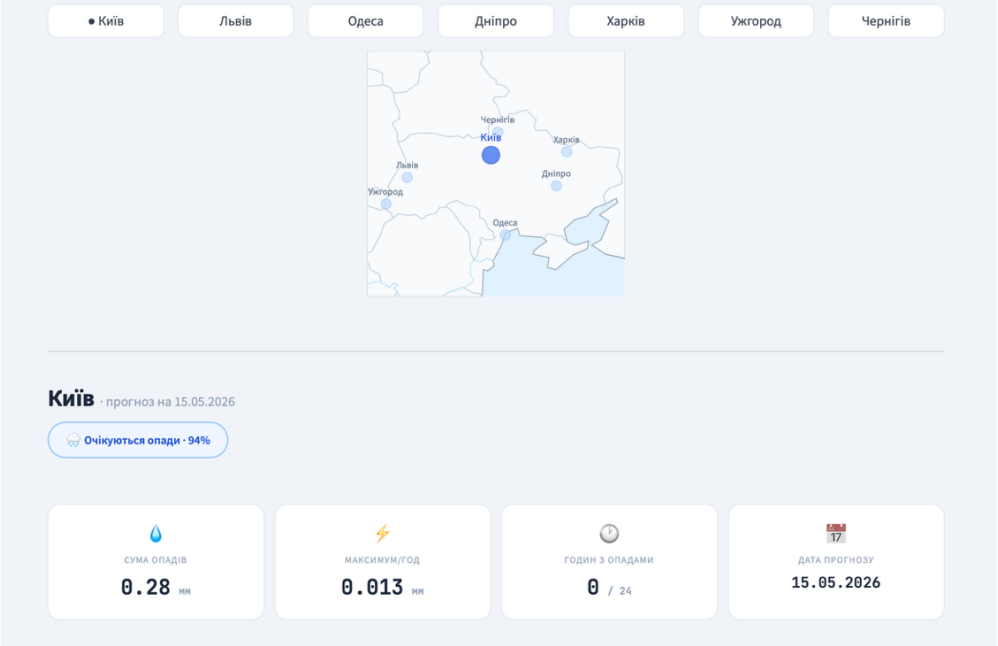

Рисунок 4.6 – Основна робоча область користувача

Інтерактивна карта з вибором міста має сім кнопок із назвами міст. Активне місто підсвічується синім і має позначку у вигляді крапки. Активне місто позначається більшою синьою крапкою і темнішим підписом. Карта є виключно візуальним елементом і не є кнопкою, а вибір міста здійснюється через кнопки або випадаючий список. Інтерактивна карта з вибором міста зображена на рисунку 4.7.

Рисунок 4.7 – Інтерактивна карта з вибором міста

Одразу під картою відображається назва обраного міста і дата прогнозу – це перша година прогнозного вектора, тобто дата на яку реально прогнозуються опади.

Над метриками знаходиться бейдж стану погоди. Він може приймати три форми. Синій бейдж «Очікуються опади» з відсотком впевненості класифікатора коли XGBoost прогнозує опади і загальна сума перевищує 0.1 мм. Жовтий бейдж «Без опадів» коли класифікатор прогнозує відсутність опадів. Зелений бейдж «Сліди опадів або відсутні» коли класифікатор може казати «буде дощ», але трансформер прогнозує загальну суму менше 0.1 мм.

Нижче розташовані чотири метричні картки. Сума опадів це загальна кількість опадів у міліметрах за всі 24 прогнозовані години, тобто сума всіх 24 значень вектора прогнозу. Максимум/год – найбільше значення з 24 годинних прогнозів, виражене з точністю до трьох знаків після коми. Годин з опадами – кількість годин із 24, де прогнозована інтенсивність перевищує поріг 0.1 мм/год. Саме цей поріг використовувався при навчанні класифікатора як межа між «є дощ» і «немає дощу». Як результат – ціле число від 0 до 24. Дата прогнозу – дата першої прогнозної години, тобто дата на яку будується прогноз. Метричні картки зображені на рисунку 4.8.

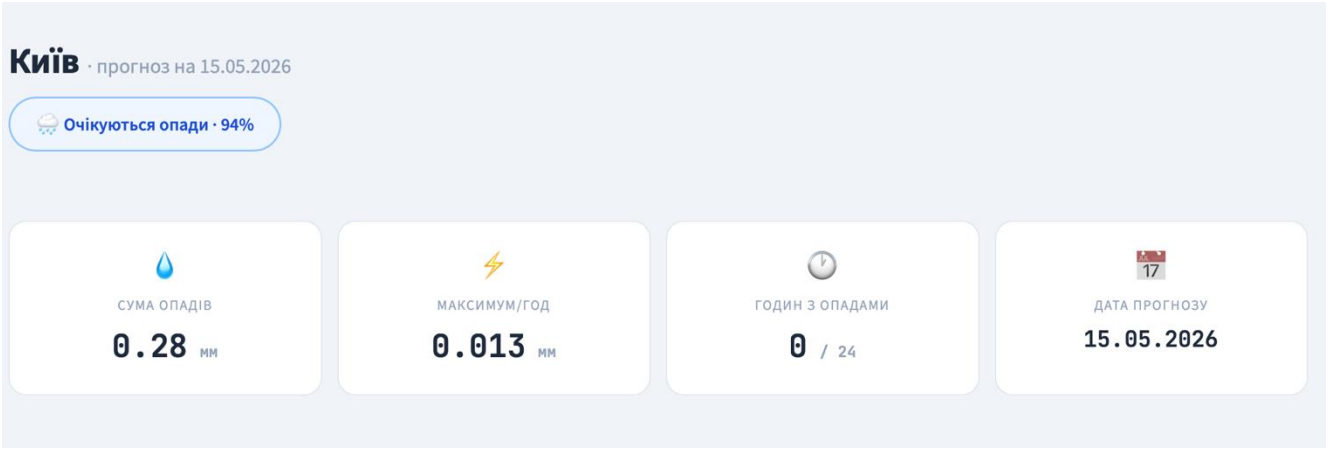

Рисунок 4.8 – Метричні картки

Перша і основна вкладка «Графіки» містить стовпчикову діаграму погодинних значень опадів з кольоровим кодуванням за інтенсивністю, кумулятивну криву накопичення опадів, кругову діаграму розподілу типів інтенсивності та теплову карту.

Перша діаграма це стовпчикова діаграма погодинних опадів. По осі X відкладено 24 часові мітки, по осі Y – інтенсивність у мм/год. Кожен стовпець розфарбований відповідно до шкали інтенсивності: світло-сірий для слідів опадів, блакитний для мікроопадів, синій для слабкого дощу, темно-синій для помірного і сильного дощу. Пунктирна горизонтальна лінія показує поріг 0.1 мм/год, а саме стовпці, що перетинають цю лінію зараховуються як «години з опадами». При наведенні курсора на будь-який стовпець з'являється підказка з точним значенням до трьох знаків після коми. Стовпчикова діаграма погодинних опадів зображена на рисунку 4.9.

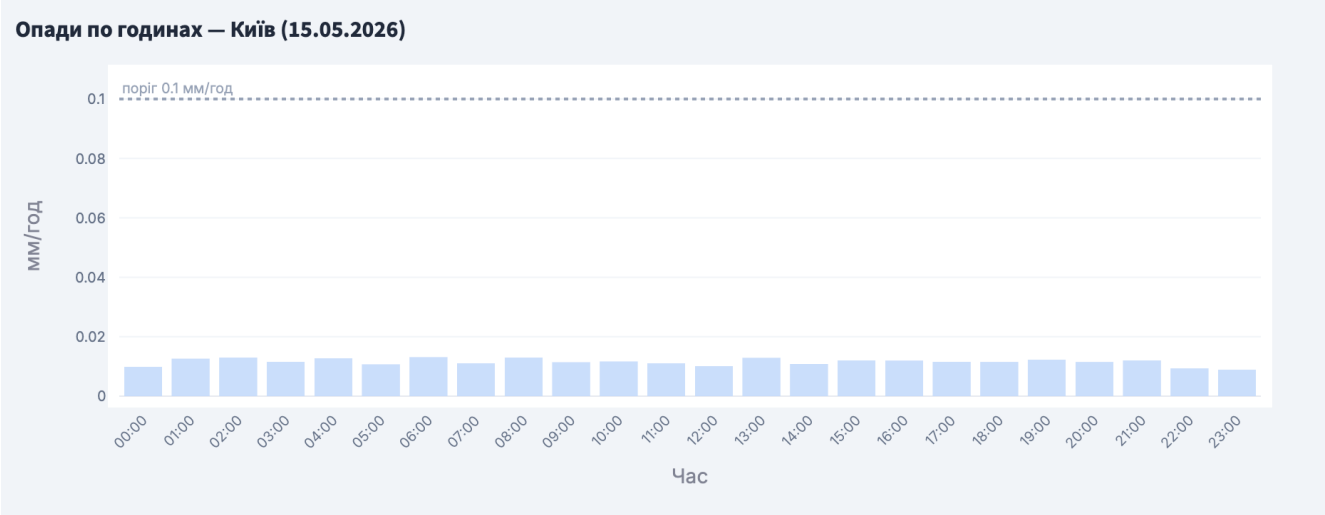

Рисунок 4.9 – Стовпчикова діаграма погодинних опадів.

Наступна візуалізація це кумулятивна крива накопичення опадів. Вона показує як накопичується загальна кількість опадів протягом доби. Якщо крива різко зростає в певний час це означає пік інтенсивності. Якщо крива пласка, то опади рівномірно розподілені або дуже слабкі. Зафарбована область під кривою

полегшує зорове сприйняття об'єму опадів. Кумулятивна крива накопичення наведена на рисунку 4.10.

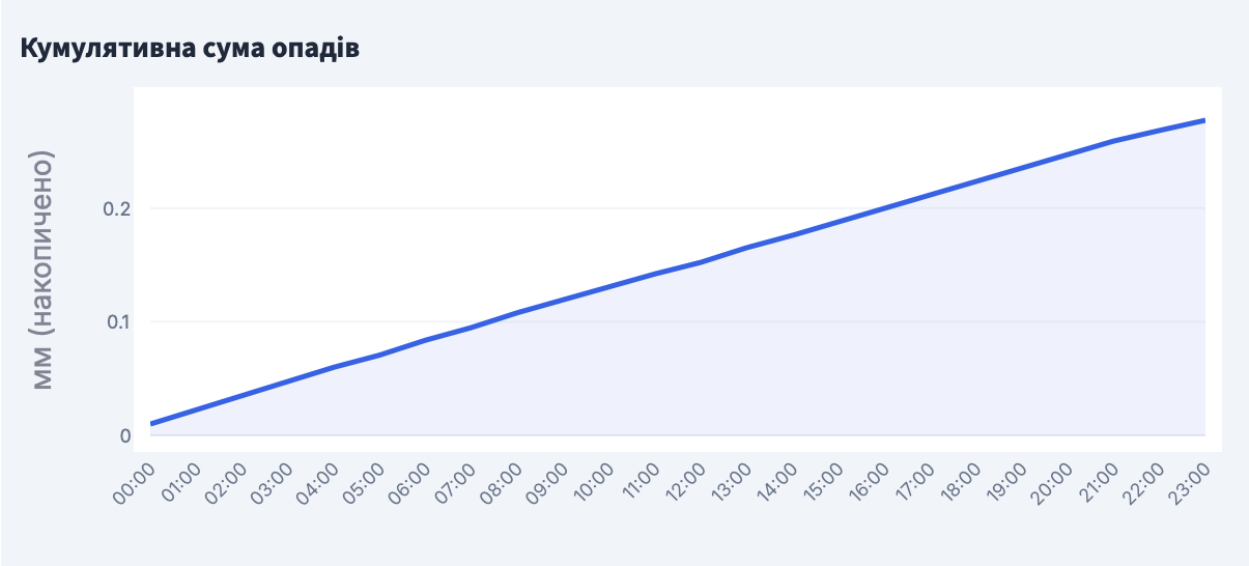

Рисунок 4.10 – Кумулятивна крива накопичення опадів

Третя візуалізація це кругова діаграма розподілу по категоріям опадів. Вона показує скільки годин із 24 припадає на кожну категорію інтенсивності. Кругова діаграма розподілу по категоріям зображена на рисунку 4.11.

Рисунок 4.11 – Кругова діаграма розподілу по категоріям опадів

Остання візуалізація це теплова карта. Вона відображає однорядковий градієнтний рядок де кожна клітинка відповідає одній годині. Колір від білого до темно-синього відображає інтенсивність. Візуалізація дозволяє миттєво побачити які години будуть найінтенсивнішими без читання числових значень. Теплову карту наведено на рисунку 4.12.

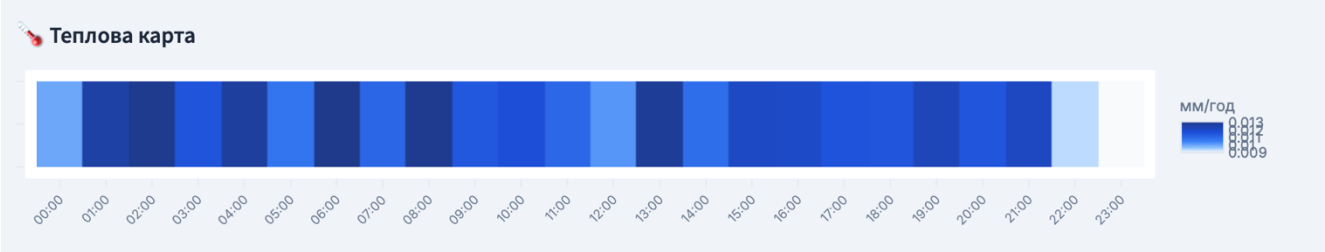

Рисунок 4.12 – Теплова карта

Наступна вкладка з робочої області – «Текстовий прогноз». Ця вкладка містить автоматично згенерований текстовий опис прогнозу. Текст формується динамічно на основі числових результатів. Вона відображає погодинний прогноз у текстовому форматі з позначенням інтенсивності, що наведено на рисунку 4.13.

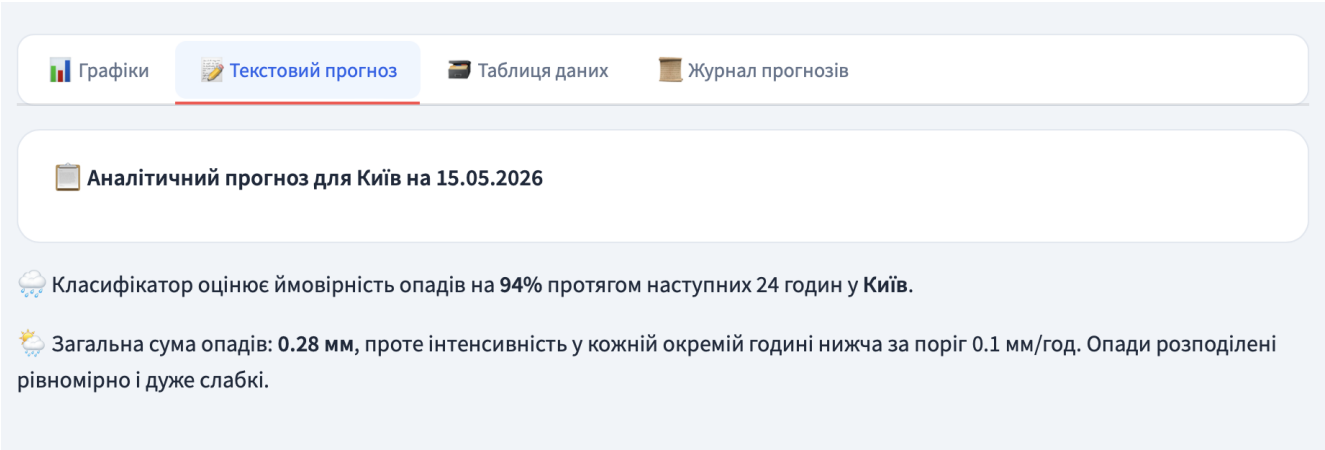

Рисунок 4.13 – Текстовий прогноз опадів

Перший абзац описує рішення класифікатора з відсотком впевненості. Другий абзац демонструє загальну характеристику, а саме якщо опадів менше 0.1 мм повідомляється, що помітних опадів не очікується, якщо є значні опади, то вказується їхня сума, максимальна інтенсивність, категорія і час піку.

Нижче розташована шкала інтенсивності опадів це довідковий елемент який пояснює всі категорії, що використовуються в системі. Кожен рядок містить кольорову крапку, назву категорії, візуальну смугу довжиною пропорційною інтенсивності і числовий діапазон у мм/год. Шкала інтенсивності прогнозованих опадів зображена на рисунку 4.14.

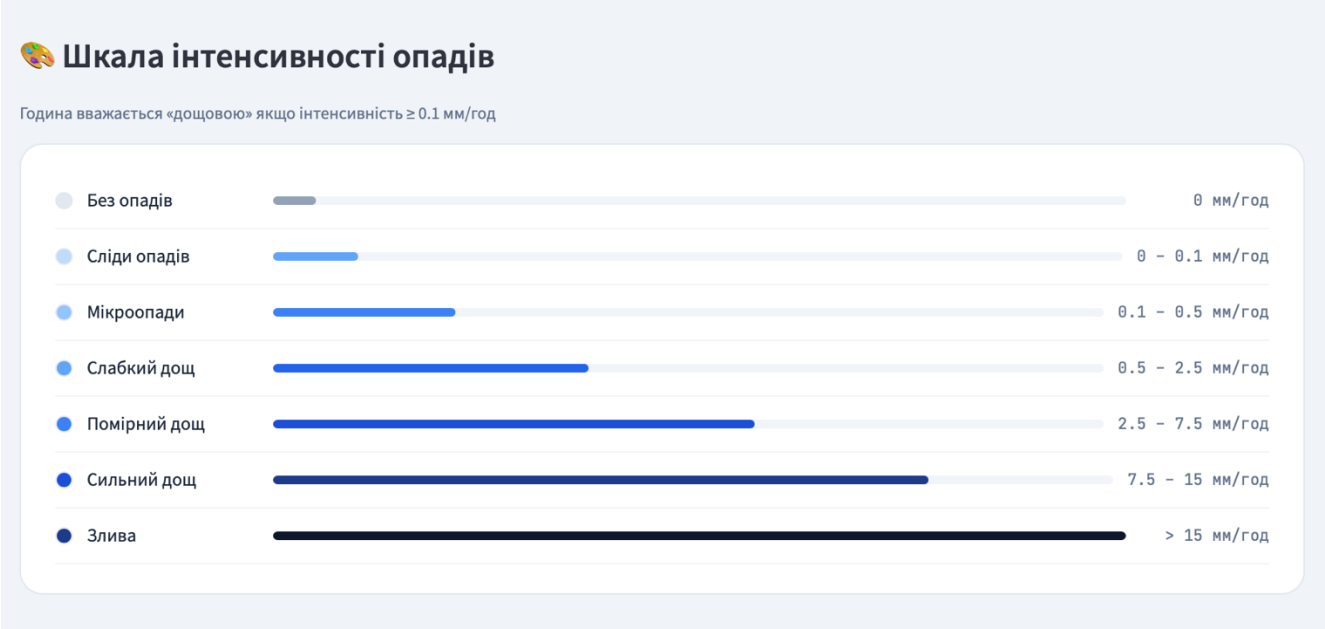

Рисунок 4.14 – Шкала інтенсивності опадів

Вкладка «Таблиця даних» відображає числові значення прогнозу у табличному форматі з можливістю експорту.

Ліва частина це погодинна таблиця з чотирма колонками: час, опади в мм/год з точністю до трьох знаків, категорія інтенсивності і накопичена сума. Таблиця прокручується і дозволяє переглянути точні числові значення для кожної години.

Права частина це зведена таблиця з ключовими показниками: дата прогнозу, загальна сума, максимум, час піку, кількість дощових годин, середнє і медіана. Під

нею кнопка завантаження CSV-файлу з повним погодинним прогнозом. Таблиці зображні на рисунку 4.15.

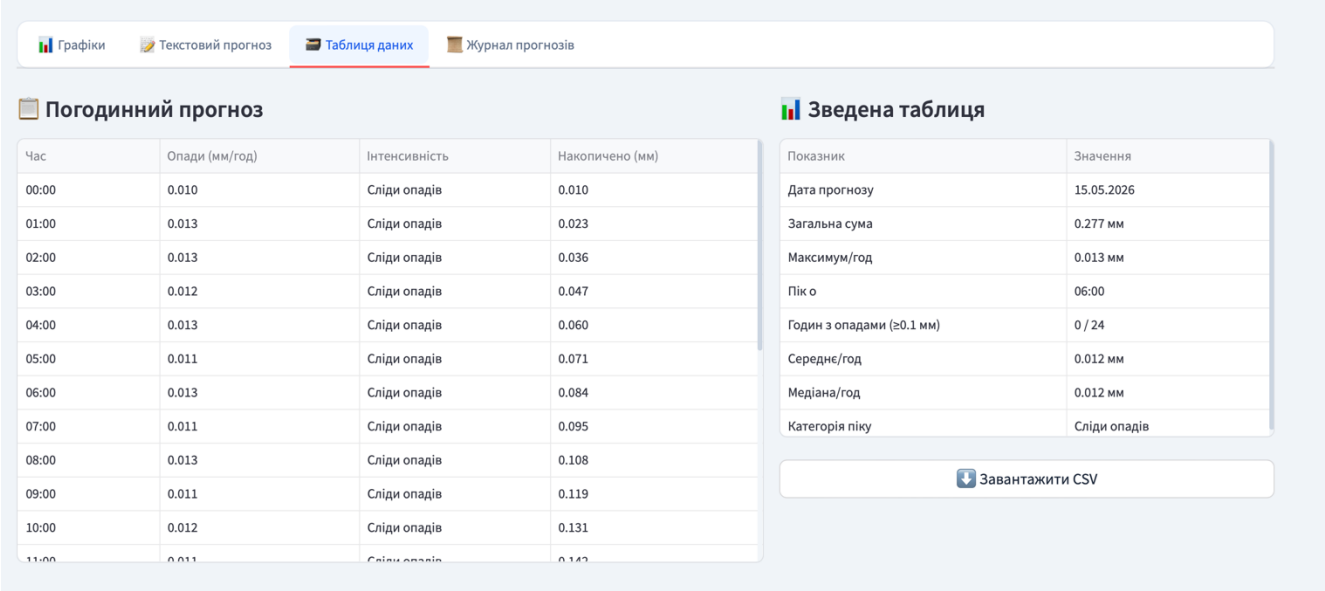

Рисунок 4.15 – Таблиці з даними опадів

Вкладка «Журнал прогнозів» зберігає і відображає історію всіх запитів та надає агреговану статистику по містах. Кожен запит прогнозу автоматично записується в JSON-файл на диску. Вкладка показує три агреговані метрики: загальну кількість запитів, найпопулярніше місто і середню суму опадів по всіх запитах.

Стрічка останніх 50 записів у зворотному хронологічному порядку демонструє час запиту, місто, сума опадів, кількість дощових годин і рішення класифікатора з відсотком впевненості. Журнал прогнозів наведений на рисунку 4.16.

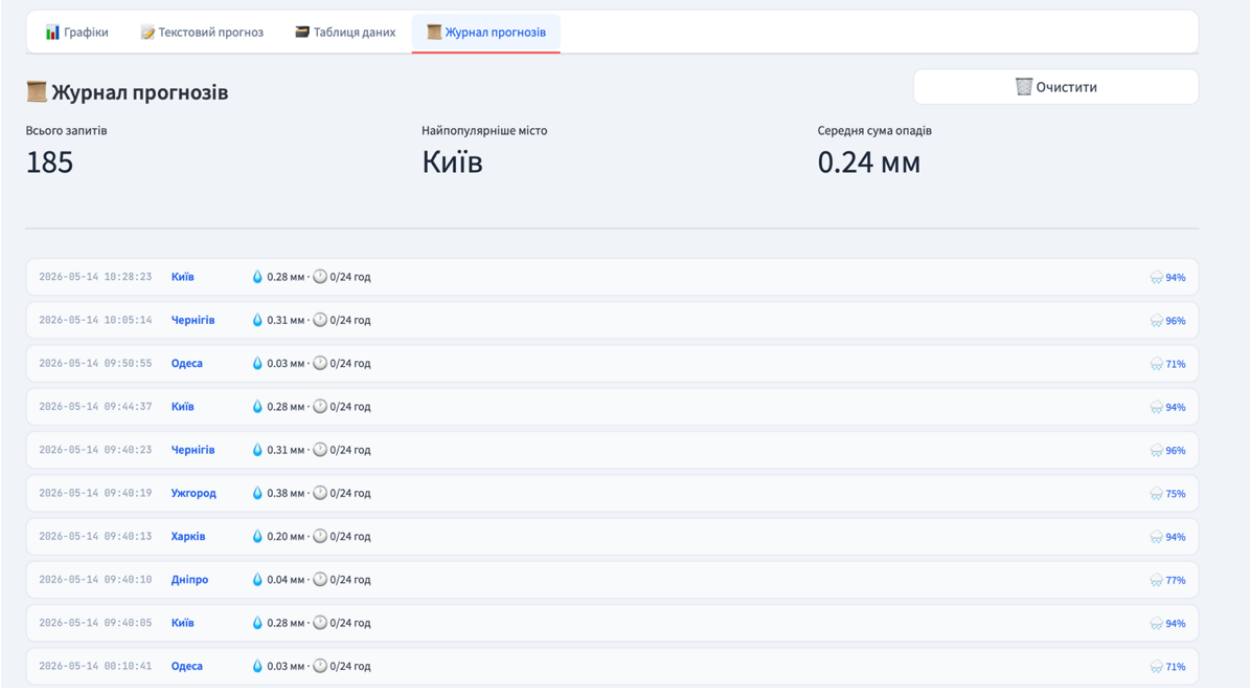

Рисунок 4.16 – Журнал прогнозів

Таким чином, інтерфейс забезпечує повний цикл взаємодії користувача з системою, від оновлення вхідних даних до отримання і аналізу результату в єдиному зручному інтерфейсі без необхідності запускати скрипти вручну через термінал.

4.3 Оцінювання якості прогнозування регресійної та класифікаційної моделей

Для задачі прогнозування погодинної інтенсивності опадів використовувалась регресійна модель Transformer. Оцінювання виконувалося за допомогою метрик RMSE, MAE, MSE та коефіцієнта детермінації R^2. У якості базового підходу розглядалася тривіальна модель, яка завжди передбачає 0 мм опадів.

У таблиці 4.2 наведено результати порівняння оцінок регресійної та базової моделей.

Таблиця 4.2 – Порівняння з базовою моделлю

Метрика	Transformer	Baseline
RMSE	0.7853	0.7698
MAE	0.2478	0.4233
MSE	0.6166	0.5926
R^2	-0.0498	-0.0088

Порівняння метрик Transformer-моделі з базовою наведено на рисунку 4.17.

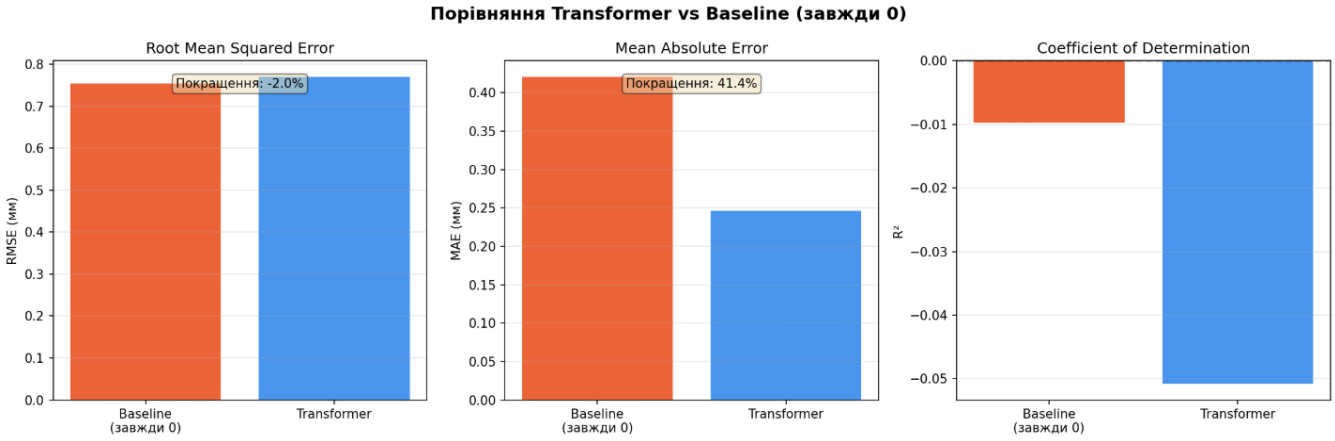

Рисунок 4.17 – Порівняння Transformer та базової моделі за метриками RMSE, MAE та R^2

Отримані результати свідчать про те, що регресійна модель демонструє ефективність у передбаченні невеликих опадів. Зокрема, значення MAE є суттєво меншим порівняно з базовою моделлю. Покращення становить 41.4%, що вказує на здатність моделі достатньо точно прогнозувати невеликі значення опадів.

Водночас значення RMSE є дещо гіршим за базову модель, що свідчить про наявність значних відхилень у прогнозах при інтенсивних опадах.

Від'ємне значення коефіцієнта детермінації R^2 є характерним для задач прогнозування опадів з нерівномірним розподілом. Причиною такої поведінки є розподіл опадів де переважна більшість годин є бездощовими. Як видно з рисунку 4.18, переважна більшість спостережень, а саме 11.5% від загальної кількості, належить до категорії слабких опадів (0.1–2.5 мм), тоді як сильні опади (10–50 мм) становлять лише 0.008% вибірки.

У таких умовах модель зміщується у бік прогнозування малих значень і не здатна ефективно відтворювати екстремальні значення, що суттєво впливає на метрику RMSE. Водночас поодинокі випадки сильних опадів генерують великі абсолютні помилки, які непропорційно впливають на метрики RMSE та R^2, чутливі до екстремальних відхилень.

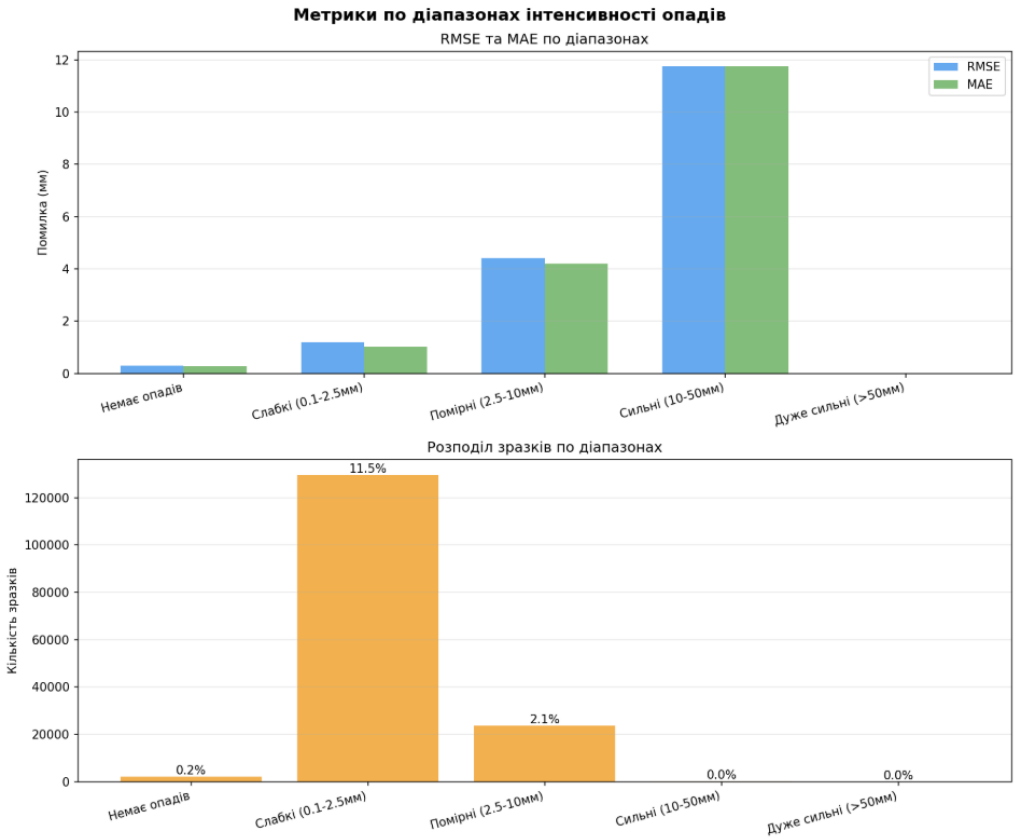

Рисунок 4.18 – Метрики RMSE та MAE по діапазонах інтенсивності опадів та розподіл вибірки

Для оцінювання задачі визначення факту наявності опадів використовувалась класифікаційна модель XGBoost. Якість класифікації оцінювалась за метриками Accuracy, Precision, Recall, F1-Score, ROC-AUC та PR-AUC. У якості базового підходу розглядалася модель, яка завжди передбачає відсутність опадів (таблиця 4.3).

Таблиця 4.3 – Метрики якості класифікаційної моделі

Метрика	XGBoost	Baseline
Accuracy	0.7358	0.5593
Precision	0.6725	0.0000
Recall	0.7807	0.0000
F1-Score	0.7226	0.0000
ROC-AUC	0.817	0.500
PR-AUC	0.788	0.441

Отримані результати демонструють, що класифікаційна модель XGBoost суттєво перевершує базовий підхід за всіма основними метриками, загальна точність зросла з 0.5593 до 0.7358. Значення Recall (0.7807) вказує на те, що модель успішно виявляє більшість випадків наявності опадів. Precision (0.6725) є дещо нижчим, що свідчить про наявність певної кількості хибних спрацювань. F1-Score на рівні 0.7226 підтверджує збалансованість між точністю та повнотою.

Детальний розподіл правильних та хибних прогнозів наведено у матриці помилок (таблиця 4.4).

Таблиця 4.4 – Матриця помилок

Факт	Прогноз: опади	Прогноз: немає опадів
Опади	15 856	4 453
Немає опадів	7 720	18 052

Отримані результати демонструють суттєву перевагу XGBoost над базовою моделлю, що наочно показано на рисунку 4.19.

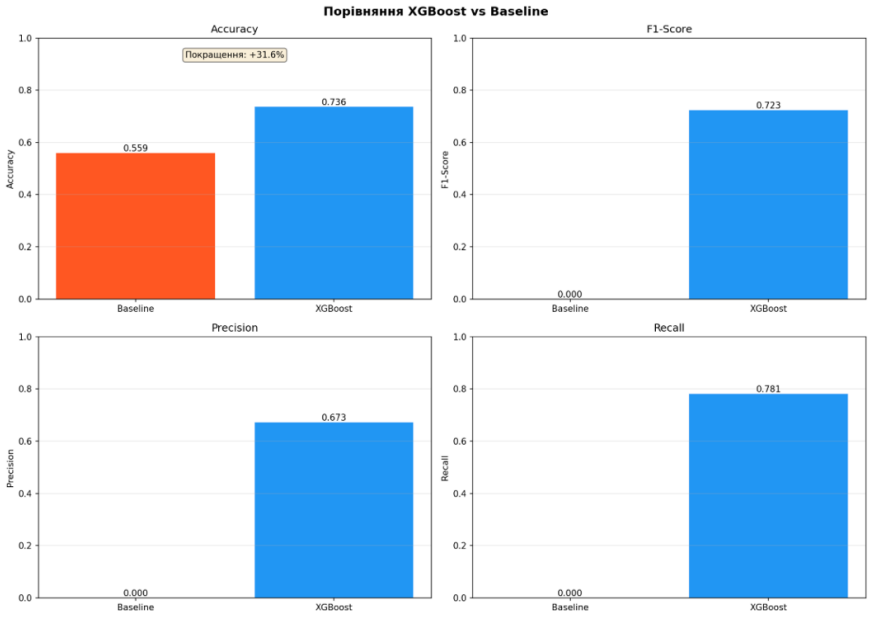

Рисунок 4.19 – Порівняння XGBoost-класифікатора та базової моделі за метриками Accuracy, F1-Score, Precision та Recall

Як видно з рисунку 4.19, XGBoost-класифікатор перевершує базову модель за всіма чотирма метриками. Особливо показовим є порівняння за F1-Score, Precision та Recall.

Базова модель отримує нульові значення за цими метриками, оскільки завжди передбачає відсутність опадів і не виявляє жодного позитивного випадку. Покращення Accuracy становить 31.6%, що підтверджує практичну ефективність навченого класифікатора.

Візуальне порівняння матриць помилок базової моделі та XGBoost-класифікатора наведено на рисунку 4.20.

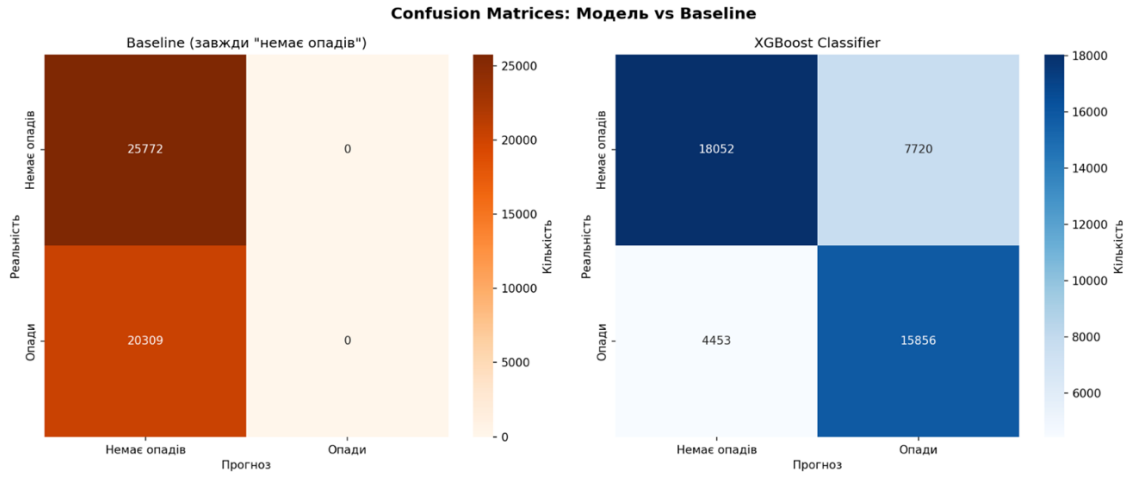

Рисунок 4.20 – Матриці помилок базової моделі та XGBoost-класифікатора

Аналіз матриці помилок показує, що модель правильно визначає як значну кількість випадків опадів, так і їх відсутність. З матриці помилок видно що модель правильно класифікувала 15856 випадків наявності опадів та 18052 випадків їх відсутності. Хибно негативних результатів – 4453, що означає пропуск реальних опадових подій. Хибно позитивних – 7720, тобто випадків коли модель передбачила опади за їх відсутності.

Для комплексної оцінки здатності моделі розділяти класи при різних порогах класифікації додатково побудовано ROC та Precision-Recall криві, наведені на рисунку 4.21.

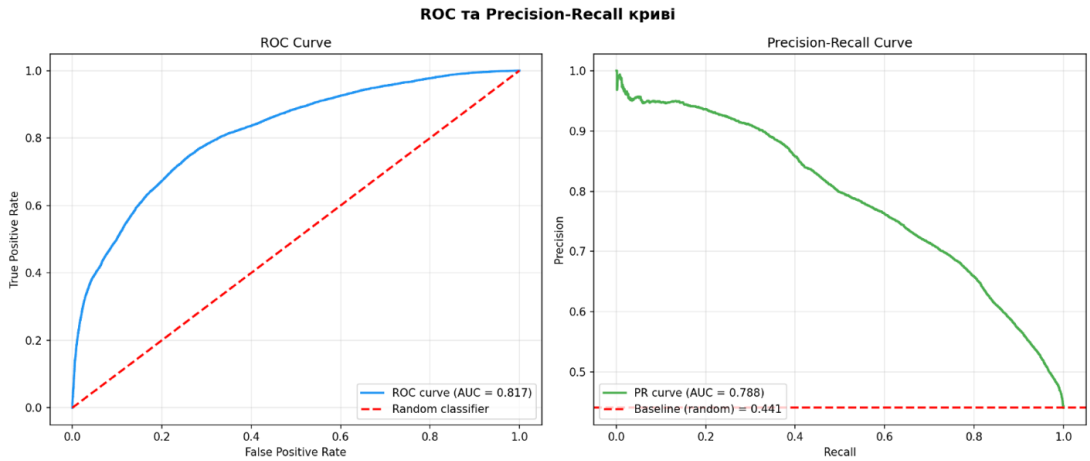

Рисунок 4.21 – ROC-крива та Precision-Recall крива

Значення ROC-AUC (0.817) та PR-AUC (0.788) свідчать про високу якість розділення класів. PR-крива значно перевищує базовий рівень (0.441), що підтверджує ефективність моделі навіть в умовах незбалансованих класів.

Спільне застосування XGBoost-класифікатора як фільтра першого рівня та Transformer-моделі для регресійного прогнозування дозволяє усунути характерну для регресійних моделей проблему хибних ненульових прогнозів у суху погоду. Такий підхід забезпечує узгодженість між якісним прогнозом наявності опадів та кількісним прогнозом їхньої інтенсивності, підвищуючи практичну корисність гібридної системи в цілому.

Результати оцінювання підтверджують ефективність обраної гібридної архітектури. XGBoost-класифікатор демонструє високу якість визначення факту наявності опадів із F1-Score 0.7226 та ROC-AUC 0.817, що суттєво перевищує базовий підхід. Transformer-модель забезпечує точне прогнозування інтенсивності малих опадів із покращенням MAE на 41.4% порівняно з базовою моделлю, однак має обмеження при прогнозуванні екстремальних значень через нерівномірний розподіл навчальних даних. Спільне застосування обох моделей у гібридній архітектурі дозволяє усунути характерну проблему хибних ненульових прогнозів у суху погоду та підвищує практичну корисність системи в цілому.

ВИСНОВКИ

У дипломній роботі розроблено програмну систему прогнозування інтенсивності атмосферних опадів для семи великих міст України із застосуванням гібридної архітектури на основі методів машинного навчання.

У ході виконання роботи проведено аналіз існуючих підходів до прогнозування атмосферних опадів. Встановлено, що традиційні чисельні методи NWP потребують значних обчислювальних ресурсів і не адаптовані до локальних кліматичних умов, тоді як методи машинного навчання, зокрема архітектура Transformer та алгоритм XGBoost демонструють конкурентоспроможну точність при значно нижчих обчислювальних витратах. Обґрунтовано доцільність гібридної двоетапної архітектури, що поєднує переваги обох підходів.

Розроблено та реалізовано програмну систему що включає п'ять функціональних модулів: збору метеорологічних даних через WeatherAPI, попередньої обробки даних, навчання моделей машинного навчання, оцінювання якості прогнозів та вебінтерфейсу користувача. Система зібрала та опрацювала п'ять років погодинних метеорологічних спостережень (березень 2021 — березень 2026) для семи міст України: Києва, Харкова, Львова, Одеси, Дніпра, Чернігова та Ужгорода.

Реалізовано Transformer-модель типу encoder-only з CLS-токеном для регресійного прогнозування погодинної інтенсивності опадів на горизонті 24 годин. Модель приймає на вхід 72-годинне вікно з 26 метеорологічними ознаками та демонструє покращення MAE на 41.4% порівняно з тривіальною базовою моделлю.

Реалізовано XGBoost-класифікатор для визначення факту наявності опадів, який досяг значень F1-Score 0.7226, ROC-AUC 0.817 та PR-AUC 0.788, що суттєво перевищує базовий підхід за всіма метриками. Класифікатор правильно виявляє 78.1% реальних опадових подій.

Реалізовано гібридний механізм узгодження прогнозів на етапі інференсу: XGBoost-класифікатор визначає факт наявності опадів, після чого Transformer формує регресійний прогноз інтенсивності, який обнуляється у разі прогнозу відсутності опадів. Такий підхід усунув характерну проблему хибних ненульових прогнозів у суху погоду.

Розроблено вебінтерфейс на основі Streamlit та Plotly що забезпечує повний цикл взаємодії з системою: від автоматичного збору даних до інтерактивної візуалізації прогнозів у вигляді погодинних діаграм, теплової карти, кумулятивної кривої та текстового аналітичного прогнозу. Система може функціонувати на персональному комп'ютері без спеціалізованого серверного обладнання.

Результати роботи підтверджують ефективність гібридної архітектури для локалізованого прогнозування атмосферних опадів і можуть бути використані в аграрному секторі, комунальних службах, системах моніторингу надзвичайних ситуацій та сервісах метеорологічного аналізу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bauer P., Thorpe A., Brunet G. The quiet revolution of numerical weather prediction. Nature. 2015. Vol. 525. P. 47–55.
2. Warner T. T. Numerical Weather and Climate Prediction. Cambridge: Cambridge University Press, 2011. 550 p.
3. Brotzge J. A. et al. Challenges and Opportunities in Numerical Weather Prediction. Bulletin of the American Meteorological Society. 2023. URL: https://doi.org/10.1175/BAMS-D-22-0172.1
4. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016. P. 785–794.
5. Frnda J., Durica M., Rozhon J., Vojtekova M., Nedoma J., Martinek R. ECMWF short-term prediction accuracy improvement by deep learning. Scientific Reports. 2022. Vol. 12. Article 7898. URL: https://www.nature.com/articles/s41598-022-11936-9
6. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997. Vol. 9, No. 8. P. 1735–1780.
7. Vaswani A. et al. Attention is all you need. Advances in Neural Information Processing Systems. 2017. Vol. 30.
8. Zhou H. et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. Proceedings of the AAAI Conference on Artificial Intelligence. 2021. Vol. 35, No. 12. P. 11106–11115.
9. Jiang M., Weng B., Chen J., Huang T., Ye F., You L. Transformer-enhanced spatiotemporal neural network for post-processing of precipitation forecasts. Journal of Hydrology. 2024. URL: https://www.sciencedirect.com/science/article/abs/pii/S0022169424001148 (дата звернення: 20.05.2026).

10. Tandon A., Awasthi A., Pattnayak K. C. Efficacy of machine learning in simulating precipitation and its extremes over the capital cities in North Indian states. Scientific Reports. 2025. Vol. 15. Article 10345. URL: https://www.nature.com/articles/s41598-024-84360-w (дата звернення: 20.05.2026).
11. Mohammed R. H., El-saieed A. M. Chaotic billiards optimized hybrid transformer and XGBoost model for robust and sustainable time series forecasting. Scientific Reports. 2025. Vol. 15. Article 25962. URL: https://www.nature.com/articles/s41598-025-10641-7 (дата звернення: 20.05.2026).
12. Törnquist E., Costa Santos W., Pogue T., Wingle N., Caulk R. A. Balancing Computational Efficiency and Forecast Error in Machine Learning-based Time-Series Forecasting: Insights from Live Experiments on Meteorological Nowcasting. arXiv. 2023. URL: https://arxiv.org/html/2309.15207 (дата звернення: 20.05.2026).
13. Subhadarsini S., Kumar D. N., Govindaraju R. S. Enhancing Hydro-climatic and land parameter forecasting using Transformer networks. Journal of Hydrology. 2025. URL: https://www.sciencedirect.com/science/article/abs/pii/S0022169425002446 (дата звернення: 20.05.2026).
14. Ramakrishnan S., Chinnappan C. C. Parallelized hybrid ensemble machine learning framework for scalable and accurate rainfall prediction. Machine Learning with Applications. 2026. Vol. 24. Article 100863. URL: https://www.sciencedirect.com/science/article/pii/S2666827026000289 (дата звернення: 20.05.2026).
15. Sun Z. Back To The Future: A Hybrid Transformer-XGBoost Model for Action-oriented Future-proofing Nowcasting. arXiv. 2024. URL: https://arxiv.org/abs/2412.19832 (дата звернення: 20.05.2026).
16. Rasp S. et al. WeatherBench: A benchmark dataset for data driven weather forecasting: вебсайт. URL: https://www.researchgate.net/publication/343929193_WeatherBench_A_Benchmark_Data_Set_for_Data-Driven_Weather_Forecasting (дата звернення: 24.05.2026).

17. Wu N., Green B., Ben X., O'Banion S. Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. arXiv. 2021. URL: https://www.semanticscholar.org/reader/f5a28db512357b700b62fb655ef4a90864e2fe7e (дата звернення: 20.05.2026).
18. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System. arXiv. 2016. URL: https://arxiv.org/pdf/1603.02754 (дата звернення: 20.05.2026).
19. Wang Y., Ni X. S. A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization. arXiv. 2019. URL: https://arxiv.org/pdf/1901.08433 (дата звернення: 20.05.2026).
20. XGBoost for Raman spectroscopy: вебсайт. URL: https://www.sciencedirect.com/science/article/abs/pii/S138614252401083 (дата звернення: 20.05.2026).
21. Extreme gradient boosting trees for profit-driven churn prediction: вебсайт. URL: https://www.sciencedirect.com/science/article/abs/pii/S0040162523006303 (дата звернення: 20.05.2026).
22. BiLSTM-I gap-filling for meteorological data: вебсайт. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8507855/ (дата звернення: 20.05.2026).
23. Don't Push the Button! Data Leakage in ML: вебсайт. URL: https://arxiv.org/html/2401.13796v2 (дата звернення: 21.05.2026).
24. Avoiding common ML pitfalls: вебсайт. URL: https://www.sciencedirect.com/science/article/pii/S2666389924001880 (дата звернення: 20.05.2026).
25. Rethinking Evaluation in Time Series: вебсайт. URL: https://arxiv.org/html/2510.13654 (дата звернення: 21.05.2026).
26. VEST: Automatic Feature Engineering for Forecasting: вебсайт. URL: https://arxiv.org/pdf/2010.07137 (дата звернення: 21.05.2026).
27. Dynamic Lagging for Time-Series Forecasting: вебсайт. URL: https://arxiv.org/html/2509.20244 (дата звернення: 21.05.2026).

28. How Much Can Time-related Features Enhance Forecasting: вебсайт. URL: https://arxiv.org/abs/2412.01557 (дата звернення: 22.05.2026).
29. Cyclical Temporal Encoding for Energy Forecasting: вебсайт. URL: https://arxiv.org/html/2512.03656v1 (дата звернення: 22.05.2026).
30. Senocak A. U. G., Yilmaz M. T., Kalkan S., Yucel I., Amjad M. An explainable two-stage machine learning approach for precipitation forecast. Journal of Hydrology. 2023. Vol. 627. URL: https://www.sciencedirect.com/science/article/abs/pii/S0022169423013173 (дата звернення: 22.05.2026).
31. Probabilistic two-stage spatial model for precipitation: вебсайт. URL: https://arxiv.org/pdf/0901.3484 (дата звернення: 22.05.2026).
32. Heydari S., Nikoo M. R., Mohammadi A., Barzegar R. Two-stage meta-ensembling machine learning model for enhanced water quality forecasting. Journal of Hydrology. 2024. Vol. 641. Article 131767. URL: https://www.sciencedirect.com/science/article/abs/pii/S0022169424011636 (дата звернення: 22.05.2026).
33. Forecast Evaluation for Data Scientists: вебсайт. URL: https://arxiv.org/pdf/2203.10716 (дата звернення: 22.05.2026).
34. Critical review on ML evaluation metrics: вебсайт. URL: https://www.sciencedirect.com/article/pii/S2213343725043714 (дата звернення: 22.05.2026).
35. Prediction Performance overview: вебсайт. URL: https://www.sciencedirect.com/topics/computer-science/prediction-performance (дата звернення: 20.05.2026).
36. Evaluation metrics in medical imaging AI: вебсайт. URL: https://www.sciencedirect.com/science/article/pii/S3050577125000283 (дата звернення: 22.05.2026).
37. ROC curve for imbalanced datasets: вебсайт. URL: https://www.sciencedirect.com/science/article/pii/S2666389924001090 (дата звернення: 22.05.2026).

38. Anomaly detection evaluation metrics: вебсайт. URL: https://arxiv.org/html/2409.15986v1 (дата звернення: 23.05.2026).
39. Confusion Matrix overview: вебсайт. URL: https://www.sciencedirect.com/topics/engineering/confusion-matrix(дата звернення: 23.05.2026).
40. Van Rossum G., Drake F. L. Python 3 Reference Manual. Scotts Valley: CreateSpace, 2009. 242 p.
41. Machine Learning in Python: вебсайт. URL: https://arxiv.org/abs/2002.04803 (дата звернення: 23.05.2026).
42. High-performance Python for Data Science and ML: вебсайт. URL: https://arxiv.org/abs/2302.03307 (дата звернення: 23.05.2026).
43. Productivity, Portability, Performance: Data-Centric Python: вебсайт. URL: https://arxiv.org/pdf/2107.00555 (дата звернення: 23.05.2026).
44. Paszke A. et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv. 2019. URL: https://arxiv.org/abs/1912.01703 (дата звернення: 23.05.2026).
45. Comparative Survey PyTorch vs TensorFlow: вебсайт. URL: https://arxiv.org/abs/2508.04035 (дата звернення: 24.05.2026).
46. Time-series data preprocessing survey: вебсайт. URL: https://www.sciencedirect.com/science/article/pii/S2307187724000452 (дата звернення: 24.05.2026).
47. Streamlit documentation: вебсайт. URL: https://docs.streamlit.io (дата звернення: 24.05.2026).

ДОДАТОК А

Реалізація моделей прогнозування інтенсивності атмосферних опадів

Програмні засоби:

– мова програмування – Python 3.12;

– середовище розробки – Visual Studio Code;

– фреймворк реалізації Transformer-моделі – PyTorch 2.9.0;

– бібліотека градієнтного бустингу – XGBoost 2.1;

– виконання операцій з масивами даних – NumPy;

– обробка датасету – Pandas.

Програмний код моделі Transformer (precipitation_transformer.py):

```
CONFIG = {
    "seq_len":       72,
    "pred_len":      24,
    "batch_size":    128,
    "d_model":       128,
    "n_heads":       8,
    "n_layers":      4,
    "d_ff":          256,
    "dropout":       0.2,
    "epochs":        100,
    "lr":            5e-5,
    "patience":      15,
    "weight_decay":  1e-4,
    "warmup_epochs": 5,
}
```

```
NON_FEATURE_COLS = {"datetime", "city", "precip_mm"}



class WeatherDataset(Dataset):
    def __init__(self, df, seq_len, pred_len):
        self.seq_len  = seq_len
        self.pred_len = pred_len
        feat_cols = [c for c in df.select_dtypes(include=[np.number]).columns
                     if c not in NON_FEATURE_COLS]
        df_feat = df[feat_cols].copy()
        df_feat = df_feat.replace([np.inf, -np.inf], np.nan).fillna(0)
        df_feat = df_feat.clip(-10, 10)
        self.features   = df_feat.values.astype(np.float32)
        self.target_reg = df["precip_mm"].values.astype(np.float32)

    def __len__(self):
        return max(0, len(self.features) - self.seq_len - self.pred_len + 1)

    def __getitem__(self, idx):
        X     = torch.tensor(self.features[idx : idx + self.seq_len])
        y_reg = torch.tensor(
            self.target_reg[idx + self.seq_len : idx + self.seq_len + self.pred_len]
        )
        return X, y_reg



class PrecipitationTransformer(nn.Module):
    def __init__(self, n_features, pred_len, cfg):
        super().__init__()
```

```
    d = cfg["d_model"]
    self.input_proj = nn.Sequential(
        nn.Linear(n_features, d),
        nn.LayerNorm(d),
        nn.ReLU(),
    )
    self.cls_token = nn.Parameter(torch.randn(1, 1, d))
    self.pos_emb   = nn.Embedding(cfg["seq_len"] + 10, d)
    enc_layer = nn.TransformerEncoderLayer(
        d_model=d, nhead=cfg["n_heads"],
        dim_feedforward=cfg["d_ff"], dropout=cfg["dropout"],
        batch_first=True, norm_first=True,
    )
    self.encoder = nn.TransformerEncoder(enc_layer, num_layers=cfg["n_layers"])
    self.regression_head = nn.Sequential(
        nn.Linear(d, d * 2), nn.GELU(), nn.Dropout(cfg["dropout"]),
        nn.Linear(d * 2, d), nn.GELU(), nn.Dropout(cfg["dropout"]),
        nn.Linear(d, pred_len),
    )
    self._init_weights()

def _init_weights(self):
    for m in self.modules():
        if isinstance(m, nn.Linear):
            nn.init.xavier_uniform_(m.weight)
            if m.bias is not None:
                nn.init.zeros_(m.bias)
    nn.init.normal_(self.cls_token, std=0.02)

def forward(self, x):
```

```
        B, T, _ = x.shape
        x   = self.input_proj(x)
        pos = torch.arange(T, device=x.device)
        x   = x + self.pos_emb(pos).unsqueeze(0)
        cls = self.cls_token.expand(B, -1, -1)
        x   = torch.cat([cls, x], dim=1)
        x   = self.encoder(x)
        return self.regression_head(x[:, 0])



class RegressionLoss(nn.Module):
    def __init__(self):
        super().__init__()
        self.mse = nn.MSELoss()
        self.mae = nn.L1Loss()

    def forward(self, pred, target):
        l_mse = self.mse(pred, target)
        l_mae = self.mae(pred, target)
        return (l_mse + l_mae) / 2, l_mse, l_mae
```

Програмний код моделі XGBoost-класифікатор (train_classifier.py):

```
CONFIG = {
    "seq_len":          72,
    "pred_len":         24,
    "rain_threshold":   0.1,
    "n_estimators":     300,
    "max_depth":        6,
    "learning_rate":    0.05,
```

```
    "subsample":        0.8,
    "colsample_bytree": 0.8,
    "min_child_weight": 5,
    "gamma":            0.1,
    "reg_alpha":        0.1,
    "reg_lambda":       1.0,
}


def prepare_features(df, seq_len, pred_len, rain_threshold):
    feat_cols = [c for c in df.select_dtypes(include=[np.number]).columns
                 if c not in NON_FEATURE_COLS]
    df_feat = df[feat_cols].copy()
    df_feat = df_feat.replace([np.inf, -np.inf], np.nan).fillna(0).clip(-10, 10)
    features = df_feat.values.astype(np.float32)
    target   = df["precip_mm"].values
    cities   = df["city"].values
    X_list, y_list = [], []

    for i in range(len(features) - seq_len - pred_len + 1):
        if cities[i] != cities[i + seq_len + pred_len - 1]:
            continue
        window = features[i : i + seq_len]
        row = np.concatenate([
            window.mean(axis=0),
            window.std(axis=0),
            window.min(axis=0),
            window.max(axis=0),
            window[-1],
            window[-3:].mean(axis=0),
```

```
            window[-6:].mean(axis=0),
            window[-12:].mean(axis=0),
        ])
        X_list.append(row)
        future = target[i + seq_len : i + seq_len + pred_len]
        y_list.append(1 if (future > rain_threshold).any() else 0)

    return np.array(X_list, dtype=np.float32), np.array(y_list, dtype=np.int32)


n_neg = (y_train == 0).sum()
n_pos = (y_train == 1).sum()
scale_pos_weight = n_neg / (n_pos + 1e-8)

model = XGBClassifier(
    n_estimators          = CONFIG["n_estimators"],
    max_depth             = CONFIG["max_depth"],
    learning_rate         = CONFIG["learning_rate"],
    subsample             = CONFIG["subsample"],
    colsample_bytree      = CONFIG["colsample_bytree"],
    min_child_weight      = CONFIG["min_child_weight"],
    gamma                 = CONFIG["gamma"],
    reg_alpha             = CONFIG["reg_alpha"],
    reg_lambda            = CONFIG["reg_lambda"],
    scale_pos_weight      = scale_pos_weight,
    eval_metric           = "logloss",
    early_stopping_rounds = 20,
    random_state          = 42,
    n_jobs                = -1,
)
model.fit(X_train, y_train, eval_set=[(X_val, y_val)], verbose=False)
```

ДОДАТОК Б

Апробація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

XXIII міжнародна науково-практична конференція молодих вчених та студентів «Сучасні проблеми наукового забезпечення енергетики:безпека, стійкість, ІТ та екологічний моніторинг»
(до 40-річчя Чорнобильської катастрофи)

м. Київ, 17–22 квітня 2026 року

Київ- 2026

UDC 004.8

[1] BS's programme 4th year Nahaivska D.O.
[1] Assist. Holovakin M.A.
https://scholar.google.com.ua/citations?hl=en&user=bdC3JMgAAAAJ

[1] Igor Sikorsky Kyiv Polytechnic Institute

APPLICATION OF THE TRANSFORMER ARCHITECTURE FOR ATMOSPHERIC PRECIPITATION INTENSITY FORECASTING

Statement of the problem and its relevance. Atmospheric precipitation significantly affects the environment, agriculture, transportation, energy systems, and construction. Inaccurate or delayed forecasting of rain, snow, or heavy showers leads to economic losses and increases the risk of floods and traffic accidents. Traditional numerical weather models provide general forecasts but are limited in the quantitative prediction accuracy under local and rapidly changing weather conditions.

Analysis of the latest research. Recent studies in meteorological forecasting increasingly focus on machine learning and deep learning methods capable of capturing complex nonlinear relationships and long-term temporal dependencies. Traditional statistical and physical weather models (Numerical Weather Prediction, NWP) effectively describe large-scale atmospheric processes but show limited accuracy in quantitative precipitation forecasting at the local level, particularly during rapid weather changes. Therefore, recent research has actively explored machine-learning algorithms to improve short-term forecasting accuracy [1, 2].

Transformer-based neural network architectures demonstrate significant advantages in tasks involving multiple interrelated time series, including weather forecasting applications [3]. In this context, parameters with complex interdependencies such as temperature, humidity, and atmospheric pressure are typically analyzed.

The use of Transformers enables modeling these interdependencies, thereby improving forecasting performance. Owing to the self-attention mechanism, Transformers effectively capture long-term dependencies in sequential data, which is particularly important for short-term precipitation intensity forecasting. Combined with optimized training algorithms such as Adam and regression loss functions including MSE (Mean Squared Error) and MAE (Mean Absolute Error), these models can achieve predictive accuracy beyond that of classical machine learning approaches.

Research objective. The objective of this study is to investigate a precipitation intensity forecasting system based on historical meteorological data using a Transformer architecture.

Main part. To forecast atmospheric precipitation intensity, we employ a model based on the Transformer architecture, as proposed in [4]. The Transformer is a neural network architecture for sequence processing, relying entirely on the attention mechanism and not using recurrent or convolutional layers. The core component of the model is the self-attention mechanism, which enables the network to determine the relative importance of each element in the input sequence when generating a forecast.

The application of the Transformer architecture is justified by the nature of meteorological data, which involve complex nonlinear interactions among multiple variables and long-term temporal dependencies. Weather processes are influenced by numerous interrelated factors; therefore, accurate forecasting requires simultaneous analysis of multiple multivariate time series. Traditional recurrent models may lose information over long sequences or require substantial computational resources.

The model input consists of a historical sequence of meteorological parameters over a specified time window, including air temperature, relative humidity, atmospheric pressure, wind

speed and direction, as well as previous precipitation values. Each time step is represented as a feature vector. The self-attention mechanism allows the model to analyze relationships between all time steps in the sequence and identify which previous atmospheric states most strongly influence future precipitation intensity. The model generates an internal representation of the temporal dynamics of meteorological parameters, which is then used to generate a forecast for the target future time horizon.

The practical advantage of using Transformer models for precipitation forecasting lies in their ability to simultaneously consider multiple meteorological variables over long time horizons, capturing subtle temporal patterns and extreme events such as heavy showers or sudden storms. This capability allows for more accurate short-term local forecasts, which can directly support decision-making in agriculture, flood management, transportation, and energy sectors. By effectively modeling the interdependencies between variables, Transformers provide a robust framework for real-time adaptive forecasting, reducing economic losses and enhancing public safety during adverse weather conditions.

The model is trained in a supervised learning setting, where each input sequence corresponds to a known target precipitation intensity value. Regression loss functions such as MSE and MAE are used to evaluate prediction errors. Model parameters are optimized using the Adam algorithm through backpropagation. An important practical aspect of implementing the proposed model is the selection of the input time window and forecasting horizon. The length of the historical sequence influences the model's ability to capture both short-term fluctuations and longer atmospheric trends. A shorter window may allow faster adaptation to rapidly changing weather conditions, while a longer window enables the model to account for accumulated atmospheric dynamics preceding precipitation events.

In addition to model design, careful data preprocessing plays a crucial role in achieving high forecasting performance. Meteorological data often contain missing values, measurement noise, and temporal inconsistencies caused by sensor errors or maintenance interruptions.

Thus, the proposed approach is based on modern deep learning techniques for time series forecasting and enables effective modeling of complex relationships among meteorological variables for improved precipitation intensity prediction.

Conclusions. It is shown that numerical and statistical models have limitations in capturing complex relationships and long-term temporal dependencies among meteorological variables. A forecasting approach based on the Transformer architecture is proposed, which leverages the self-attention mechanism to effectively analyze interrelated time series data. The model structure, training procedure, and optimization methods are described. The implementation of the proposed approach is expected to improve short-term precipitation intensity forecasting accuracy and enable local predictions.

References:

1. Zhang T., et al. *Deep learning of flood forecasting*. 2025. URL : https://hess.copernicus.org/articles/29/5955/2025/index.html (date of access: 26.02.2026)

2. Rasp S., Dueben P.D., Scher S., Weyn J.A., Mouatadid S., Thuerey N. *WeatherBench: A benchmark dataset for data driven weather forecasting*. 2020. URL : https://doi.org/10.48550/arXiv.2002.00469 (date of access: 26.02.2026)

3. Wu N., Green B., Ben X., O'Banion S. *Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case* [Electronic resource]. 2021. URL: https://doi.org/10.48550/arXiv.2001.08317 (date of access: 26.02.2026)

4. Vaswani A., Shazeer N., Parmar N. *Attention Is All You Need*. 2017. URL: https://doi.org/10.48550/arXiv.1706.03762 (date of access: 26.02.2026)